



Jose Ignacio González Gómez.
Departamento de Economía Financiera y Contabilidad
Universidad de La Laguna
WWW.JGGOMEZ.EU
www.ecofin.ull.es/users/jggomez.

Tema:

Apuntes Programación Excel VBA.

PARTE III

(Mis Rutinas y trucos a aplicar)

Indice

1 PROTECCIÓN Y DESPROTECCIÓN DE LIBROS.

- 1.1 PROPUESTA MÁS RECIENTES PARA DESPROTEGER UN EXCEL
- 1.2 PROTEGER UN LIBRO, EL CÓDIGO VBA Y UNA HOJA CONCRETA.
- 1.3 DESPROTEGER UN PROYECTO VBA POR QUE NOS HEMOS OLVIDADO LA CLAVE.
- 1.4 DESPROTEGER SOLO LAS HOJAS DE UN LIBRO PORQUE NOS HEMOS OLVIDADO DE LA CLAVE.
 - 1.4.1 Propuesta reciente
- 1.5 OTRA ALTERNATIVA QUE ME FUNCIONO CON PACIENCIA DEJANDO QUE TRABAJE EN EXCEL 2007. (JUNIO 2010)
- 1.6 OTRA ALTERNATIVA SIMILAR A LA ANTERIOR
- 1.7 DESPROTEGER UN MODULO VBA FUERTEMENTE PROTEGIDO

2 BLOQUEAR LA EJECUCIÓN DE UNA MACRO CON CLAVE. O ACCESO A UN APARTADO CON CLAVE

- 2.1 PLANTEAMIENTO DEL PROBLEMA
- 2.2 ALTERNATIVA 1. INCLUYENDO EL CÓDIGO EN LA MACRO.
- 2.3 ALTERNATIVA 2. INCLUYENDO EL CÓDIGO EN LA MACRO Y USANDO UN USERFORM.
- 2.4 MI PROPUESTA.
 - 2.4.1 Introducción. Creación del modulodeclaves.
 - 2.4.2 Programación del botón enviar del formulario clave.
 - 2.4.3 Aplicación a un botón.

3 DISEÑO BÁSICO DE LA ESTRUCTURA DE UNA APLICACIÓN EN EXCEL.

- 3.1 INTRODUCCIÓN.
- 3.2 CONFIGURANDO LA HOJA DE PRESENTACIÓN DE NUESTRA APLICACIÓN.
 - 3.2.1 Proteger la hoja de inicio
 - 3.2.2 Quitando la referencia A, B, C, D..... 1,2 3
 - 3.2.3 Quitando la barra inferior con el nombre de las hojas.
- 3.3 OCULTAR HOJAS DE UN LIBRO.
 - 3.3.1 Planteamiento del problema.
- 3.4 LIMITAR EL DESPLAZAMIENTO EN UNA HOJA

4 VINCULAR TABLAS Y GRÁFICOS DE EXCEL EN WORD

- 4.1 PLANTEAMIENTO DEL PROBLEMA
- 4.2 PREPARAR EL DOCUMENTO. CÓMO ESTABLECER UNA BASE DE HIPERVÍNCULO EN EL WORD
- 4.3 PROBLEMA DEL OFFICE CON CONVERSIÓN.

5 HACER UN SISTEMA DE AYUDA PARA NUESTRA APLICACIÓN.

- 5.1 PREPARATIVOS DEL FICHERO AYUDA EN WORD Y GENERACIÓN DE LA WEB.
- 5.2 CREACIÓN DE NUESTRO PROYECTO DE FORMA SIMPLE.
 - 5.2.1 Generación del fichero hhp.
 - 5.2.2 Configurar las ventanas, crear la ventana Main
 - 5.2.3 Compilando el fichero hhp, generando el fichero CHM Simple.
- 5.3 MEJORANDO Y AMPLIANDO NUESTRO PROYECTO.
 - 5.3.1 Crear el contenido de la ayuda. Lista de Tópicos. Fichero TOC.
 - 5.3.2 Crear el índice de la ayuda. Fichero Index.
- 5.4 MODIFICAR NUESTRO PROYECTO. CAMBIAR LAS OPCIONES DEL PROYECTO.
- 5.5 DEFINIR LOS TÓPICOS PARA USAR DESDE NUESTRA APLICACIÓN. CONFIGURAR LOS ALIAS DE LA AYUDA
- 5.6 PROGRAMANDO EL BOTÓN QUE LLAME AL FICHERO DE AYUDA CHM.

6 UNA CINTA DE OPCIONES PERSONALIZADA. MENÚ RIBBONX.

- 6.1 OBJETIVO. CREAR UNA BARRA DE MENÚ PERSONALIZADA.
- 6.2 ASPECTOS GENERALES DE LA NUEVA BARRA DE HERRAMIENTAS DEL OFFICE 2007.
 - 6.2.1 La Cinta de opciones Ribbon.
 - 6.2.2 El formato XML de la cinta de opciones Ribbon.
 - 6.2.3 Programas de apoyo para el desarrollo de la cinta de opciones. Ribbon Creator y Custom UI Editor.
- 6.3 PROGRAMA RIBBON CREATOR.
 - 6.3.1 Creación de un Nuevo Proyecto Ribbon.
 - 6.3.2 Crear un cinta de opciones con Ribbon Creator para aplicarla a un fichero Excel existente.
 - 6.3.3 Vincular el complemento Excel a nuestra hoja de calculo personalizada.

7 OTRAS CUESTIONES

- 7.1 INSERTAR IMÁGENES EN EXCEL COMO ICONOS, APLICAR PARA MENÚ
- 7.2 HACER REFERENCIA A UNA RUTA RELATIVA EN EXCEL.
- 7.3 ABRIR UN DOCUMENTO WORD DESDE EXCEL Y SITUARLO EN UNA PÁGINA CONCRETA.
- 7.4 ¿PUEDO HACER QUE UNA O VARIAS CELDAS DEL LIBRO PARPADEEN?
- 7.5 MACROS AUTO_OPEN() Y AUTO_CLOSE().WORKBOOK_OPEN

8 BIBLIOGRAFÍA.

1 Protección y desprotección de libros.

1.1 Propuesta más recientes para desproteger un EXCEL

Fuente: <http://vbaparaaplicacioneseexcel.blogspot.com.es/2013/01/eliminar-la-proteccion-en-libros-y-hojas.html>
07/01/2015

ELIMINAR LA PROTECCION EN LIBROS Y HOJAS

Sé que existen miles de publicaciones al respecto pero siempre viene bien tenerlas a mano y entender cómo funciona.

Estos dos procedimientos son muy útiles cuando recibimos algún archivo protegido, y no nos comunican la contraseña para poder abrirlo. Sea como fuere, olvido o necesidad, cualquier problema de este tipo podemos solucionarlo gracias a estos dos procedimientos;

Para quitar la protección a un libro utilizaremos este procedimiento:

```
Sub DesprotegerLibro()
If MsgBox("Realmente desea desproteger el libro actual?", _
vbCritical + vbYesNo + vbDefaultButton2, "DesprotegerLibro") = vbYes
Then
Dim i As Integer, j As Integer, k As Integer
Dim l As Integer, m As Integer, n As Integer
On Error Resume Next
For i = 65 To 66: For j = 65 To 66: For k = 65 To 66
For l = 65 To 66: For m = 65 To 66: For i1 = 65 To 66
For i2 = 65 To 66: For i3 = 65 To 66: For i4 = 65 To 66
For i5 = 65 To 66: For i6 = 65 To 66: For n = 32 To 126
ActiveWorkbook.Unprotect Chr(i) & Chr(j) & Chr(k) & Chr(l) & Chr(m) &
Chr(i1) & Chr(i2) & Chr(i3) & Chr(i4) & Chr(i5) & Chr(i6) & Chr(n)
If ActiveWorkbook.ProtectStructure = False And
ActiveWorkbook.ProtectWindows = False Then
MsgBox "El libro está ahora desprotegido", vbInformation + vbOKOnly,
"DesprotegerLibro"
Exit Sub
End If
Next: Next: Next: Next: Next: Next
Next: Next: Next: Next: Next: Next
End If
End Sub
```

1.2 Proteger un libro, el código VBA y una hoja concreta.

El password inicial, es decir la clave para abrir el libro es bastante sencillo; en lugar de elegir Guardar, debemos elegir Guardar Como... y allí hay un botón que dice Herramientas y luego Opciones Generales, en este momento podemos elegir la contraseña ya sea de apertura como de solo lectura.

Respecto a la opción de proteger el código VBA, desde el entorno VBA debemos seleccionar el proyecto y presionar botón derecho con el mouse veremos una opción que es Propiedades del VBAProject y ahí seleccionamos la solapa Protección y tildas la opción que dice Bloquear proyecto para visualización y le colocamos un password.

Respecto a la protección de una hoja/s concreta/s es necesario hacerlo a través de una macro. Las instrucciones son:

ActiveSheet.Protect Password:="micontraseña"

ActiveSheet.Unprotect Password:="micontraseña"

Si queremos proteger sin contraseñas será:

Activesheet.protect

1.3 Desproteger un proyecto VBA por que nos hemos olvidado la clave.

En caso de que nos hemos olvidado de la clave de un proyecto VBA lo mejor es utilizar el programa Advanced VBA Password Recovery en su versión 1.6.

Pero a veces no es suficiente, por ello he encontrado una solución que funciona al 100% para los ficheros que se ponen pesados. Reproduzco a continuación lo extraído de:

http://foros.emagister.com/mensaje-truco_para_quitar_contraseña_a_proyectos_en_vba_de_excel-12890-684259-1-2402753.htm

- Paso 1: Haz clic en inicio y en ejecutar teclea la siguientes letras: cmd .
PRESIONA ENTER. Se abrirá una ventana del Símbolo de sistema
- Paso 2: Luego escribe o teclea lo siguiente: edit /70 PRESIONA ENTER . Se abrirá la ventana con menú del sistema.
- Paso 3: Desde esa venta selecciona tu archivo y lo abres. Nota importante: haz una copia del archivo original para mayor seguridad.
- Paso 4: En el menú BUSCAR, busca el texto "DPB", asegúrate que existe inmediatamente atrás de las letras "GC=" y [Host Extender Info]
- Paso 5: Al texto "DPB" cámbialo por la siguiente escritura: "DBx"
- Paso 6: Selecciona Guardar y guarda el archivo el Archivo
- Paso 7: Ahora abre el archivo en Excel recibirás un mensaje de error, ignóralo y selecciona Si (Yes) o Aceptar

En este paso aun no podrás acceder al proyecto VBA

- Paso 8: Abre el Editor de Visual Basic (Herramientas - Macro - Editor de Visual Basic)
- Paso 9: Abre las propiedades del proyecto eligiendo el menú Herramientas y luego la opción - Propiedades de VBA Project.
Selecciona la pestaña "Protección". Ingresa una contraseña cualquiera y anótala
- Paso 10: Confírmala y pulsa OK ó bien en Aceptar. (La ventana se cerrará).

- Paso 11: Después vuelve a abrir Propiedades de VBA Project ubicado en el Menú Herramientas.

Allí desactivarás la casilla "Bloquear proyecto para la visualización" y borrarás TODOS los caracteres en la que hayas escrito en la contraseña así como en confirmar contraseña.

- Paso 12: Finalmente Guarda el proyecto (Boton Guardar); al abrirlo de nuevo

¡Listo tendrás tu archivo para poder ver el proyecto completo!

1.4 Desproteger solo las hojas de un libro porque nos hemos olvidado de la clave.

1.4.1 Propuesta reciente

A veces resulta difícil desproteger una hoja concreta, para ello insertamos la siguiente macro PasswordBreaker en la hoja y nos desactivara la protección

http://groups.google.es/group/microsoft_public_es.excel/browse_thread/thread/6368dda9b881088a/34aa25425dc1b4bf?hl=es&lnk=st&q=desproteger+una+hoja#34aa25425dc1b4bf

```
Sub PasswordBreaker()
    'Author unknown
    'Breaks worksheet password protection.
    Dim i As Integer, j As Integer, k As Integer
    Dim l As Integer, m As Integer, n As Integer
    Dim i1 As Integer, i2 As Integer, i3 As Integer
    Dim i4 As Integer, i5 As Integer, i6 As Integer
    On Error Resume Next
    For i = 65 To 66: For j = 65 To 66: For k = 65 To 66
    For l = 65 To 66: For m = 65 To 66: For i1 = 65 To 66
    For i2 = 65 To 66: For i3 = 65 To 66: For i4 = 65 To 66
    For i5 = 65 To 66: For i6 = 65 To 66: For n = 32 To 126
    ActiveSheet.Unprotect Chr(i) & Chr(j) & Chr(k) & _
        Chr(l) & Chr(m) & Chr(i1) & Chr(i2) & Chr(i3) & _
        Chr(i4) & Chr(i5) & Chr(i6) & Chr(n)
    If ActiveSheet.ProtectContents = False Then
        MsgBox "One usable password is " & Chr(i) & Chr(j) & _
            Chr(k) & Chr(l) & Chr(m) & Chr(i1) & Chr(i2) & _
            Chr(i3) & Chr(i4) & Chr(i5) & Chr(i6) & Chr(n)
    Exit Sub
    End If
    Next: Next: Next: Next: Next: Next
    Next: Next: Next: Next: Next: Next
End Sub
```

1.5 Otra alternativa que me funciona con paciencia dejando que trabaje en Excel 2007. (Junio 2010)

¿Cómo descubrir la contraseña que protege una hoja de cálculo?.

Truco válido para desvelarnos el password de una hoja de cálculo protegida (no de un libro). Válido -en principio-, para todas las versiones de Excel. Extraído de la revista Computer Hoy.

Pasos a seguir:

1. Crea un fichero nuevo de Excel (ej: Libro1.xls).

2. Pulsa Alt y F11, para entrar en el proyecto VBA (Visual Basic para Aplicaciones).
3. En el menú de opciones, selecciona "Insertar". Selecciona a continuación "Módulo".
4. En la parte izquierda, donde se observa el Proyecto, se habrá creado una carpeta llamada "Módulos", y dentro de ella, figurará "Módulo1".
5. Nos situaremos en la ventana de la derecha al clicar sobre de ese "Módulo1", y copiaremos este código:

Código:

```
Sub Descubrir_contraseña()
Dim a As Integer, b As Integer, c As Integer
Dim d As Integer, e As Integer, f As Integer
Dim a1 As Integer, a2 As Integer, a3 As Integer
Dim a4 As Integer, a5 As Integer, a6 As Integer
On Error Resume Next
For a = 65 To 66: For b = 65 To 66: For c = 65 To 66
For d = 65 To 66: For e = 65 To 66: For a1 = 65 To 66
For a2 = 65 To 66: For a3 = 65 To 66: For a4 = 65 To 66
For a5 = 65 To 66: For a6 = 65 To 66: For f = 32 To 126
Contraseña = Chr(a) & Chr(b) & Chr(c) & Chr(d) & Chr(e) & Chr(a1) _
& Chr(a2) & Chr(a3) & Chr(a4) & Chr(a5) & Chr(a6) & Chr(f)
ActiveSheet.Unprotect Contraseña
If ActiveSheet.ProtectContents = False Then
MsgBox "¡Enorabuena!" & vbCr & "La contraseña es:" & vbCr & Contraseña
Exit Sub
End If
Next: Next: Next: Next: Next: Next
Next: Next: Next: Next: Next: Next
End Sub
```

6. A continuación, una vez hayamos pegado ese código, seleccionaremos en el menú de opciones, "Archivo", y a continuación "Exportar archivo". Le pondremos de nombre Descubrir_contraseña.bas (por ejemplo, aunque puede ser otra, ya que no tiene porqué tener el mismo nombre que el procedimiento cuyo código aparece en el punto 5 anterior), y lo guardaremos por ejemplo, en "Mis documentos".
7. Ahora imaginemos que tenemos una hoja de cálculo protegida, y que queremos descubrir su password. Para poder saber cual es su password, tendremos que acceder al proyecto (Alt+F11), e importar el módulo (.bas) anterior. Para ello, desde el menú de opciones, seleccionamos "Archivo", y a continuación "Importar archivo". Buscamos la ruta donde habíamos guardado el fichero del punto 6 anterior, y lo seleccionamos. Salimos ahora del proyecyo VBA, para volver a la hoja de cálculo protegida.
8. Con la hoja de cálculo cuya contraseña deseamos descubrir, en primer plano (activa), seleccionamos desde Excel (no desde VBA, aunque también se puede hacer, pero lo haremos de la forma más gráfica y sencilla posible), en el menú de opciones "Herramientas". A continuación, seleccionamos "Macro" y luego "Macros..." (si quereis ahorraros eso, pulsad Alt+F8). En la lista de macros, seleccionamos el que se llama "Descubrir_contraseña", y pulsamos el botón "Ejecutar".
9. En ese momento, comienza el procedimiento para obtener el password de la hoja. El proceso puede tardar desde unos cuantos segundos, hasta algo más de un minuto (o incluso varios minutos). No os preocupeis, que el sistema no ha entrado en un bucle infinito , tan solo hay que tener un poco de paciencia.

10. Para finalizar, el sistema nos mostrará un mensaje, diciéndonos cuál era el password de la hoja, a la vez que la misma se desprotege. Puede que esa contraseña no sea literalmente la que puso el autor de la hoja que hasta hace unos segundos estaba protegida, pero tanto el password original, como este nuevo que el sistema nos da, son equivalentes. Por ejemplo, si tenemos una hoja protegida con el password "pepe", la podemos desproteger con ese password, o con este que es el que obtenemos al ejecutar ese macro: "AAABBABBBAA/". Podeis probar que tras proteger una hoja con el password "pepe", ésta se puede desproteger con el password "pepe" (evidentemente), y también con el password "AAABBABBBAA/".

1.6 Otra alternativa similar a la anterior

http://www.lawebdelprogramador.com/foros/Ofimatica/699666-Desproteger_libro_Excel.html

Desbloquear y despoteger un archivo de Excel

Para desbloquear y desproteger una hoja de cálculo de Excel si nos olvidamos la contraseña, lo vamos a hacer mediante una rutina que te vamos a proveer a continuación, sigue estas instrucciones:

1. Abre normalmente el archivo Excel que quieres desbloquear o desproteger.
2. Pulsa ALT + F8 y se te abrirá una ventana; en el nombre del macro pones "gensktruco - Excel" o un nombre a tu elección, y pulsa el botón Crear.
3. Se te abrirá una nueva ventana, allí sustituyes y borras todo lo que esta escrito y lo reemplazas por el siguiente código:

Sub breakit()

```
Dim i As Integer, j As Integer, k As Integer
Dim l As Integer, m As Integer, n As Integer
On Error Resume Next
For i = 65 To 66
For j = 65 To 66
For k = 65 To 66
For l = 65 To 66
For m = 65 To 66
For i1 = 65 To 66
For i2 = 65 To 66
For i3 = 65 To 66
For i4 = 65 To 66
For i5 = 65 To 66
For i6 = 65 To 66
For n = 32 To 126
ActiveSheet.Unprotect Chr(i) & Chr(j) & Chr(k) & _
Chr(l) & Chr(m) & Chr(i1) & Chr(i2) & Chr(i3) & _
Chr(i4) & Chr(i5) & Chr(i6) & Chr(n)
If ActiveSheet.ProtectContents = False Then
MsgBox "La contraseña es: " & Chr(i) & Chr(j) & _
Chr(k) & Chr(l) & Chr(m) & Chr(i1) & Chr(i2) & Chr(i3) _
```


Paso 10:

Confírmalo con OK (Aceptar)

Paso 11:

Después repite estos pasos y desactiva la casilla "Bloquear proyecto para la visualización" y borra los caracteres en la contraseña así como en confirmar contraseña.

Paso 12:

Guarda el proyecto (Botón Guardar)

Listo ! Ya tienes tu archivo para poder ver el proyecto completo.

2 Bloquear la ejecución de una macro con clave. O Acceso a un apartado con clave

2.1 Planteamiento del problema

En síntesis el problema se resume en bloquear una macro de manera que si se introduce el código esta ejecute y en caso contrario salga un mensaje de texto de advertencia y salga de la ejecución de la misma. Un ejemplo sería que tenemos un formulario con un botón que da acceso a la hoja2 que por defecto esta oculta. Si ejecutamos el botón este nos pedirá una clave si esta es correcta se muestra la hoja 2 y al salir de la citada hoja se vuelve a ocultar, pero en el caso de que la clave no sea la correcta nos debería aparecer el mensaje "clave incorrecta, no tiene autorización".

2.2 Alternativa 1. Incluyendo el código en la macro.

En la macro que queremos proteger incluimos el siguiente código:

```
con = InputBox("Escribe la contraseña?", "Contraseña")
If con = "123456" Then
    ' AQUÍ ESCRIBE TODO EL CÓDIGO DE LA MACRO
Else
    MsgBox "La contraseña no es correcta", vbInformation, "Error"
End If
```

En este caso mediante un inputbox se pide que ponga una contraseña y la guarda en la variable con. Luego mira si esta es igual a la contraseña correcta, (En este caso le he puesto jggomez) Si lo es, ejecuta el código y si no sale un mensaje de error.

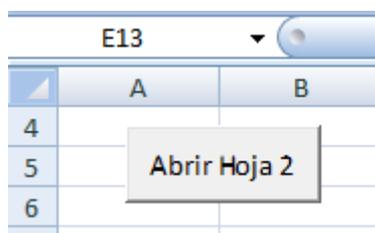


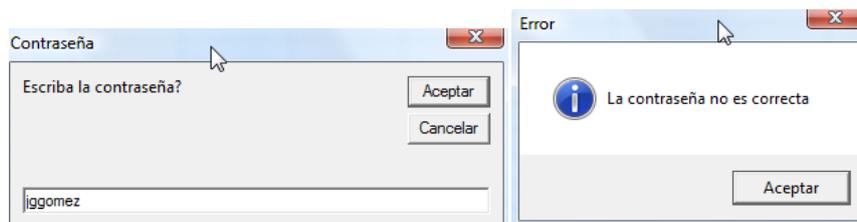
Ilustración 1

Y el código del botón será:

```

CommandButton1 Click
Private Sub CommandButton1_Click()
' Pedir clave primero antes de ejecutar la macro
con = InputBox("Escriba la contraseña?", "Contraseña")
If con = "jggomez" Then
' Aqui escribimos el código que nos interesa
' En concreto que muestre la hoja 2 y sea la activa
Worksheets(2).Visible = True
Worksheets(2).Activate
Else
MsgBox "La contraseña no es correcta", vbInformation, "Error"
End If
End Sub

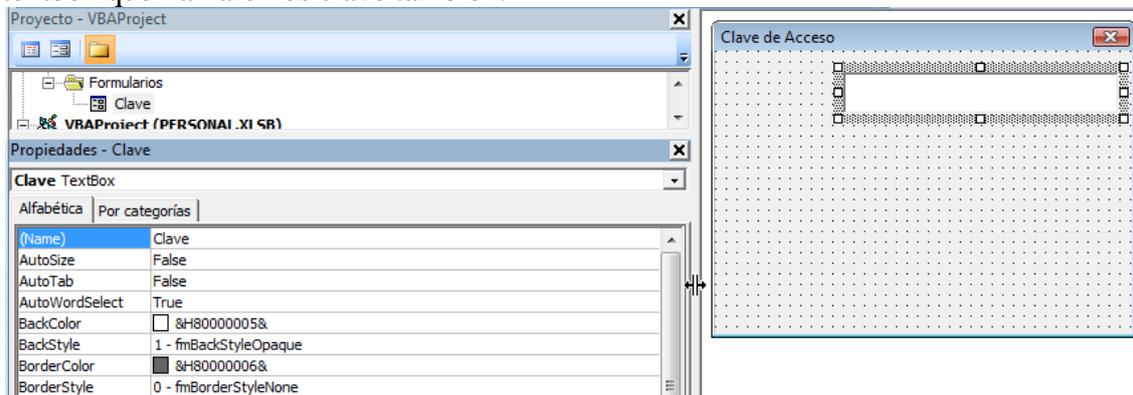
```



2.3 Alternativa 2. Incluyendo el código en la macro y usando un Userform.

En este caso y para evitar que me salga claramente la contraseña vamos a utilizar un userform.

Para ello primeramente creamos un Userform que llamaremos clave con un solo campo textbox que llamaremos clave también.



Posteriormente seleccionamos el input de texto, a la izquierda, en la ventana propiedades, bajamos hasta la opción PasswordChar, e introducimos un asterisco (*), ...aunque si prefieres otro carácter para ocultar el password, puedes poner cualquiera, pero siempre queda más elegante el asterisco.

Para el tema de la validación, y comprobar si el password introducido es correcto, debes poner un botón de aceptar, y que al hacer clic creas un procedimiento que contenga un if parecido a este:

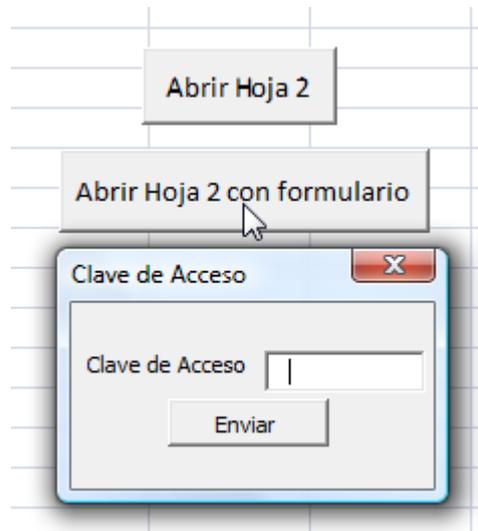
```

sub clic_en_aceptar()
'suponiendo que no le hayas cambiado el nombre al input, y que el password
'correcto sea "jggomez" (sin las comillas)...
if TextBox1="pepe"
'aquí lo que quieras hacer si el password introducido es pepe (el correcto)
else
'aquí lo que tiene que hacer si el password no es correcto

```

```
end if
end sub
```

Es decir, tendríamos un nuevo botón como vemos en la ilustración que nos abriría el formulario para entrar la clave, en este caso el código del botón sería:



```
Private Sub CommandButton2_Click()
Clave.Show
End Sub
```

Es decir mostraría el formulario llamado Clave para que cuando se haga clic se muestre. Ahora vamos a ver el código del botón enviar del formulario:

```
Private Sub Enviar_Click()
'suponiendo que no le hayas cambiado el nombre al input
If Clave = "jggomez" Then
    'aquí lo que quieras hacer si el password es correcto
    Worksheets(2).Visible = True
    Worksheets(2).Activate
Else
    'aquí lo que tiene que hacer si el password no es correcto
    MsgBox "La contraseña no es correcta", vbInformation, "Error"
End If
End Sub
```

2.4 Mi Propuesta.

2.4.1 Introducción. Creación del modulodeclaves.

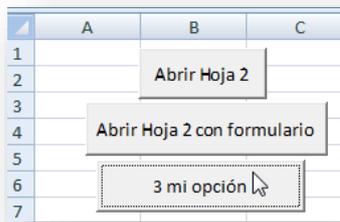


Ilustración 2

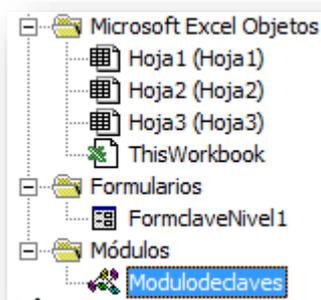


Ilustración 3

Vamos a analizar la propuesta que hemos desarrollado para dar respuesta al problema planteado, nos basaremos en un formulario que hemos llamado FormclaveNivel1 y que nos pedirá la clave de acceso y por otro un modulo que hemos llamado Modulodeclaves.

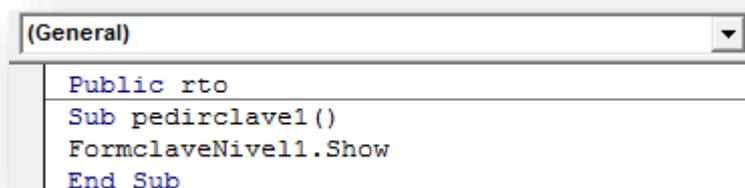
En primer lugar creamos el formulario de forma sencilla y cuyo aspecto será el siguiente:



Ilustración 4

Al inputbox lo llamamos “Clave”, es ahí donde el usuario introducirá la clave y también observamos el botón enviar que programaremos posteriormente.

El módulo “Módulodeclaves” tiene como objetivo agrupar una serie de procedimientos que nos permitirá abrir los distintos formularios para introducir las claves, en nuestro caso contamos tenemos el procedimiento “pedirclave1” que lo único que hace es abrir el formulario anterior(FormclaveNivel1). Así este modulo contiene las siguientes líneas de programación:



Señalar que en este modulo hemos declarado una variable de carácter global y por tanto estará disponible para todos los procedimientos inscritos dentro de este modulo que hemos llamado rto y cuyo objetivo es darle un valor “ok” si la clave introducida es correcta o un valor “ko” si la clave introducida es incorrecta.

Bien ya tenemos el formulario donde introduciremos la clave y el modulo anterior que utilizaremos para que nos abra el formulario y además nos guarde el resultado de si la clave esta bien introducida o no, de esta forma invocado en cualquier macro o procedimiento la siguiente sentencia: Modulodeclaves.pedirclave1 nos creara la variable publica rto y además nos abrirá el formulario.

2.4.2 Programación del botón enviar del formulario clave.

Vamos a programar a continuación el botón enviar del formulario. Recordemos que el TextBox del citado formulario lo hemos llamado Clave y que una de sus propiedades PasswordChar la hemos configurado con asterisco para que no sea vea la clave introducida.

PasswordChar * 

Bien la programación del botón enviar es la siguiente:

```

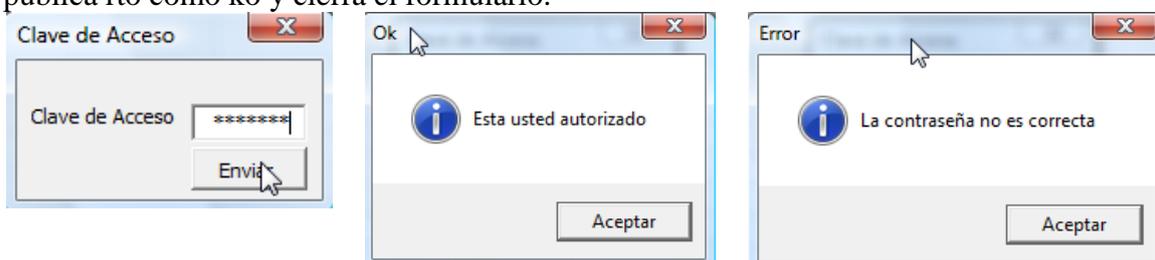
Enviar Click
Private Sub Enviar_Click()
'Establecemos el valor de la clave que debe ser la correcta
If Clave = "jggomez" Then
'aquí lo que quieras hacer si el password es correcto
MsgBox "Esta usted autorizado", vbInformation, "Ok"
'asigno valor a la variable
rto = "ok"
'cierra el formulario
Unload FormclaveNivel1
Else
'aquí lo que tiene que hacer si el password no es correcto
MsgBox "La contraseña no es correcta", vbInformation, "Error"
' asigno valor a la variable
rto = "ko"
'cierro el formulario
Unload FormclaveNivel1
End If
End Sub

```

Significa que una vez el usuario a introducido la clave en el TextBox - Clave y apreta el evento click del botón enviar sucede lo siguiente:

Definimos la clave que consideramos correcta para este nivel de seguridad, en nuestro ejemplo jggomez y si esta es igual al valor que hemos introducido en el TextBox - Clave Entonces nos muestra un mensaje en formulario tipo vbinformation, ok y además asigna el valor a la variable publica rto como ok y cierra el formulario.

En caso que no sea correcto el valor introducido en el TextBox - Clave nos muestra un mensaje de advertencia tipo vbinformation, error y además asigna el valor a la variable publica rto como ko y cierra el formulario.



Bien de esta forma tenemos definido correctamente nuestro sistema ahora simplemente lo podremos aplicar a los distintos eventos o macros que deseamos, así por ejemplo vamos programar el botón 3 que tenemos en la hoja de cálculo ver Ilustración 2.

2.4.3 Aplicación a un botón.

Explicaremos el objetivo del citado botón, este es simple, la hoja 2 se encuentra oculta, queremos que si el usuario tiene autorización pueda activar la citada hoja2 y poner esta como activa en ese momento, en caso contrario que el usuario no tenga autorización no le deje ejecutar la macro.

Para ello vamos a exponer el procedimiento asociado al evento click del boton3

```

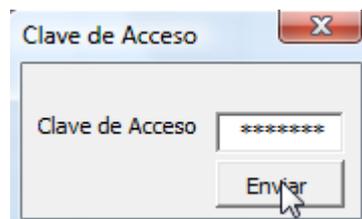
Bton3 Click
Private Sub Bton3_Click()
Modulodeclaves.pedirclave1
If rto = "ok" Then
'Muestra la hoja 2 y que sea la activa
MsgBox "chachi, entra tio"
Worksheets(2).Visible = True
Worksheets(2).Activate
Else
'En caso que clave no sea correcta
MsgBox "No tienes el nivel de autorización", vbInformation, "Error"
End If
End Sub

```

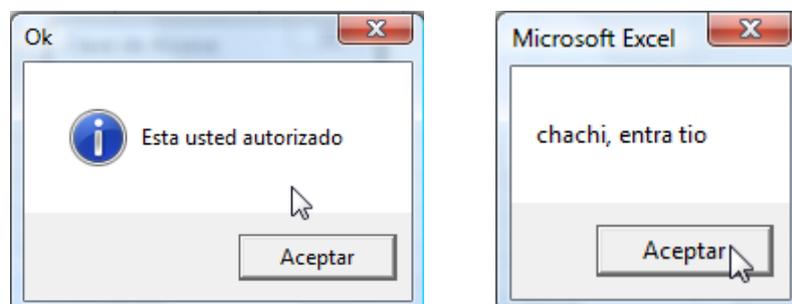
Ilustración 5

Simplemente lo que hace es:

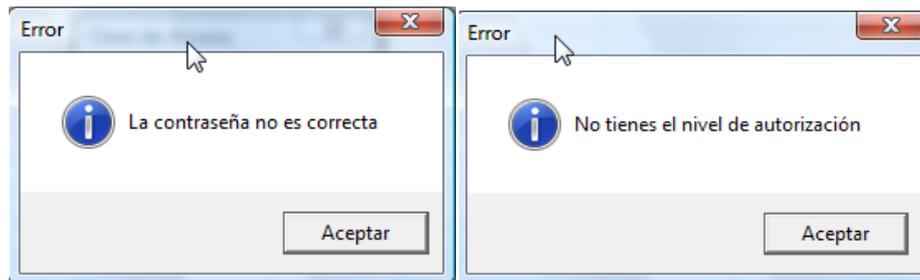
- Carga el formulario a través de la ejecución del Modulodeclaves.pedirclave1, con ello se crea la variable publica rto y se abre el formulario anterior (FormclaveNivel1) el cual nos solicitara la clave.



- Si la clave ha sido correcta, la variable rto toma el valor ok, entonces muestra el mensaje de entrada y procede a poner visible la hoja 2 y activarla.



- Si la clave no ha sido correcta, la variable rto toma el valor ko, entonces muestra el mensaje de error.



En resumen, de esta forma modificando las líneas del botón 3 y adaptándola a las macros podemos tener distintos niveles de seguridad en nuestra aplicación.

3 Diseño básico de la estructura de una aplicación en Excel.

3.1 Introducción.

En este apartado queremos exponer los aspectos básicos que pueden ayudarnos a diseñar una aplicación Excel con aspecto más o menos profesional.

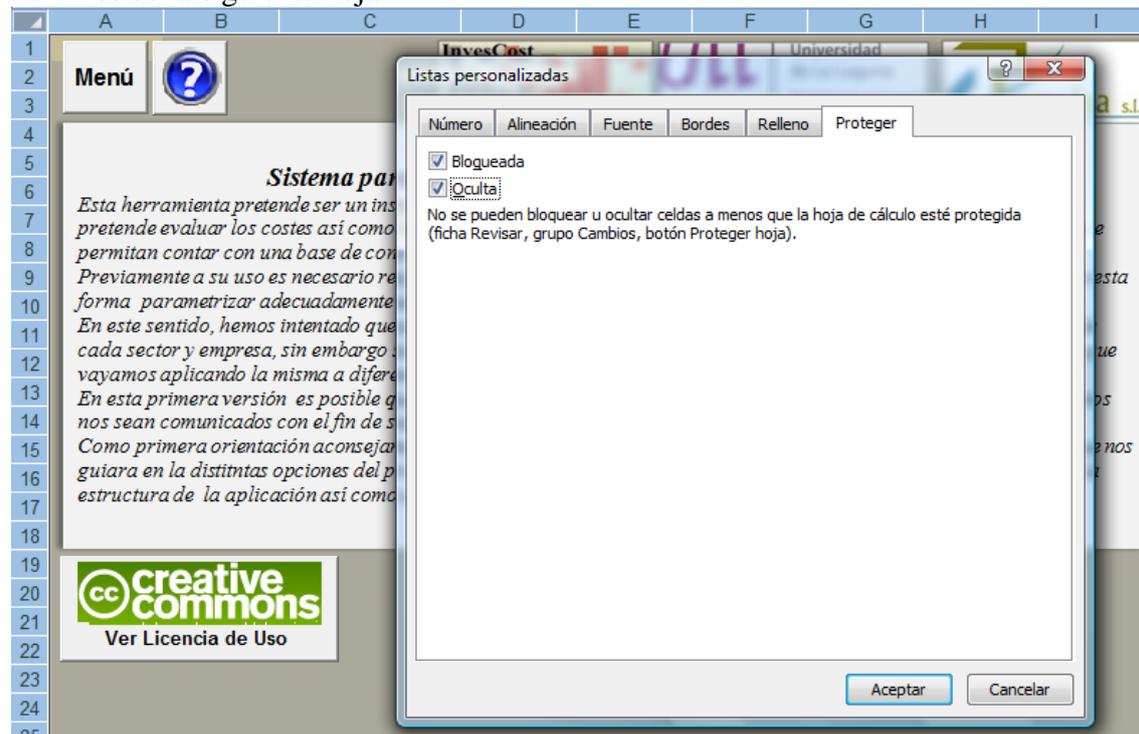
La idea central es que cada vez que abramos nuestra hoja de cálculo, se nos presente un menú navegable a través del cual nos desplazamos por las diversas opciones de la aplicación, ello supone contar con una pantalla de inicio o bien a través de un formulario y con las hojas de la aplicación oculta y mostrarlas o bien ocultarlas a medida que la necesitamos.

3.2 Configurando la hoja de presentación de nuestra aplicación.

3.2.1 Proteger la hoja de inicio

En este caso, contamos con una primera hoja de presentación que queremos proteger para que no sea modificada ni alternada y no se pueda navegar por ella.

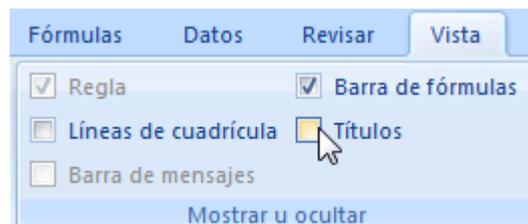
Partimos de la siguiente hoja



Seleccionamos todas las celdas y las bloqueamos y ocultamos. A continuación protegemos el libro, así solo se podrán usar los botones de navegación habilitados.

3.2.2 Quitando la referencia A, B, C, D..... 1,2 3

Para darle un aspecto más profesional a la pantalla de presentación, nos interesa quitarle las referencias de filas y columnas tradicionales de Excel para ello tenemos que desactivar los títulos de la hoja como se muestra en el menú Vista.



Si quisiéramos realizarlo a nivel de programación la instrucción sería:

ActiveWindow.DisplayHeadings = False

De esta forma nos quedaría algo así:

```
Workbook      Open
Private Sub Workbook_Open()
Application.Caption = "(c) José Ignacio González Gómez y Sandra M
Worksheets("Inicio").Select
ActiveWindow.DisplayHeadings = False
Licencial.Show
End Sub
```

Ilustración 6

Al abrir el libro, se nos carga la hoja inicio pero le quitamos los títulos. Además se carga un formulario que hemos llamado Licencial

3.2.3 Quitando la barra inferior con el nombre de las hojas.

Pretendemos quitar la barra de pestaña inferior donde aparece el nombre de todas las hojas de cálculo que nos permite navegar es decir donde están todas las hojas abiertas, para ello tenemos que activar que al cargar el libro poner la siguiente instrucción lo que supondría cambiar la Ilustración 6:

ActiveWindow.DisplayWorkbookTabs = False

3.3 Ocultar hojas de un libro.

3.3.1 Planteamiento del problema.

Deseamos que en un determinado libro solo este visible la hoja índice y la hoja que vamos requiriendo y cuando esta no sea la hoja activa ocultarla.

Cómo ocultar una hoja de cálculo Excel ofrece la posibilidad de ocultar una hoja de cálculo para, por ejemplo, preservar datos que son utilizados por otras hojas del mismo libro pero que no interesa que sean vistos o modificados por terceras personas. Esto se consigue fácilmente seleccionando en el menú Formato » Hoja » Ocultar.

El problema está en que las hojas ocultas pueden volver a visualizarse sin ningún problema mediante el comando Formato » Hoja » Mostrar. Es cierto que para impedir la modificación de toda la hoja o de parte de la misma se puede utilizar el menú Herramientas » Proteger, pero este sistema, aunque puede ocultar la visualización de las fórmulas, sigue mostrando el contenido de la hoja. El truco que proponemos se basa en la modificación directa de la propiedad Visible del objeto WorkSheet, o lo que es lo mismo, del objeto hoja de cálculo. Veamos cómo:

1. Pulse la combinación de teclas ALT + F11 para acceder al Editor de Visual Basic.
2. En la ventana del Explorador de Proyectos (si está oculta pulse CTRL + R) localice el nombre del proyecto correspondiente a su libro, por ejemplo, VBAProject (Prueba.xls) y haga clic en el nombre de la hoja que quiere ocultar.
3. En la ventana Propiedades (pulse la tecla F4 si está oculta) localice la propiedad Visible y cambie su valor eligiendo xlSheetVeryHidden en la lista desplegable. No se preocupe si al seleccionar este valor la propiedad sigue mostrando el valor xlSheetVisible.

Antes de cerrar el Editor conviene que proteja estos cambios con una contraseña de tal modo que se impida la modificación no autorizada de la propiedad Visible. Para ello sólo tiene que acceder al menú Herramientas » Propiedades de VBAProject, en la solapa Protección marcar la opción Bloquear proyecto para visualización e introducir la contraseña en el apartado Contraseña para ver las propiedades del proyecto. Una vez protegido el proyecto, salga del editor pulsando la combinación de teclas ALT + Q y guarde los cambios en su libro. Obviamente, para ver la página oculta tendrá que asignar el valor xlSheetVisible a la propiedad Visible.

<http://www.idg.es/pcworld/Excel-a-todas-potencia/art170070.htm>

3.4 Limitar el desplazamiento en una hoja

Las hojas de cálculo disponen de un número elevado de filas y columnas, en el caso de Excel 65.536 filas y 256 columnas. Hay veces que interesa limitar el área de desplazamiento de tal forma que sólo se pueda visualizar un rango determinado de filas y columnas.

Piense que, de esta manera, se podría impedir el acceso a ciertas zonas de la hoja de cálculo o sencillamente hacer más cómoda la tarea de seleccionar celdas.

Para limitar el desplazamiento es necesario modificar una propiedad de la hoja de cálculo Scroll-Area. Localice esta propiedad en la hoja que le interese y asígnele como valor el rango de celdas visible, por ejemplo, \$A\$1:\$J\$33. Con esto conseguirá que no se pueda acceder más allá de la columna J y de la fila 33.

<http://www.idg.es/pcworld/Excel-a-todas-potencia/art170070.htm>

4 Vincular tablas y gráficos de Excel en Word

4.1 Planteamiento del problema

En Excel tenemos creado una serie de tablas y gráficos que queremos insertar en Word, para ello debemos copiar los datos de Excel y posteriormente vincularlos de tal forma que nos permita actualizar cada vez que los datos originales cambien.

El problema que se plantea es que el vínculo que se genera tiene una referencia absoluta y no relativa, por tanto si cambiamos de directorio, esto no funcionara, es decir no se actualizan los vínculos.

El caso concreto es que si el documento Word que contiene el informe esta en el directorio C:\basura\Informe.doc y la hoja de calculo esta en el mismo sitio C:\basura\Datos.xls. si cambiamos de directorio estos vínculos se romperán.

Por tanto debemos establecer los vínculos como relativos.

4.2 Preparar el documento. Cómo establecer una base de hipervínculo en el Word

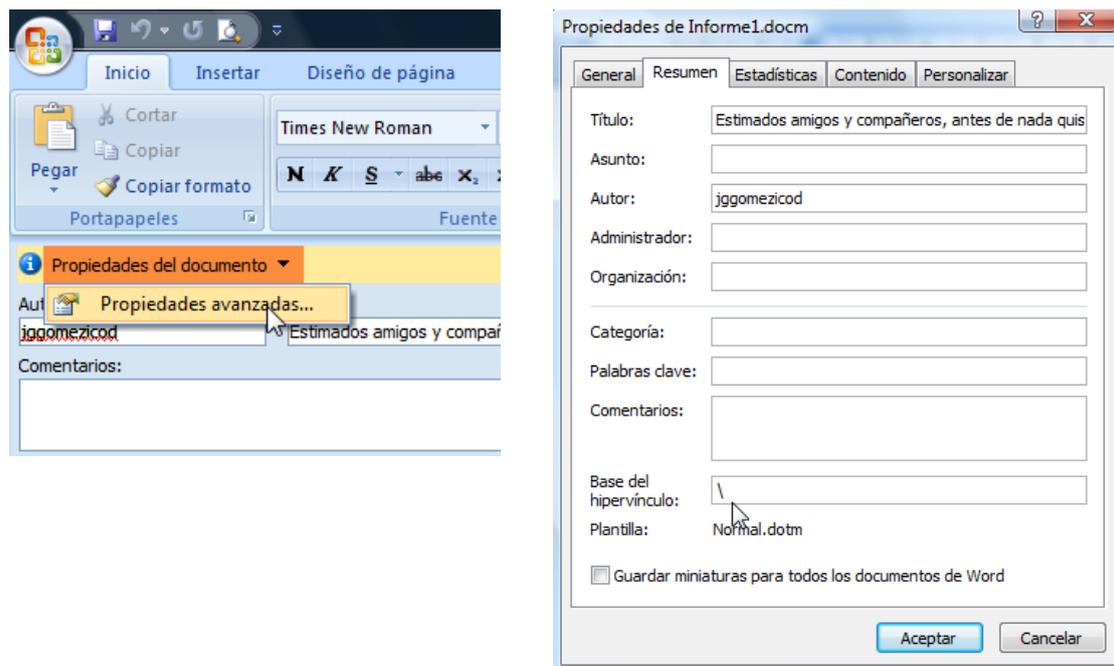
Para establecer una base de hipervínculo para todos los hipervínculos o las direcciones URL de un documento, utilice uno de los métodos siguientes según su versión de Word.

Método 1: Word 2007

1. Abra el documento en el que desee establecer una base de hipervínculo.
2. Haga clic en el botón de Microsoft Office, seleccione Preparar y haga clic en Propiedades.
3. Haga clic en Propiedades del documento y, a continuación, haga clic en Propiedades avanzadas.
4. Haga clic en la ficha Resumen.
5. En el cuadro Base de hipervínculo, escriba la ruta de acceso que desee utilizar para todos los hipervínculos que cree en este documento.
6. Haga clic en Aceptar.

Así por ejemplo vamos a tomar un documento de texto y vamos a establecer un hipervínculo con ámbito relativo, es decir asociado a los documentos que se encuentra en ese lugar.



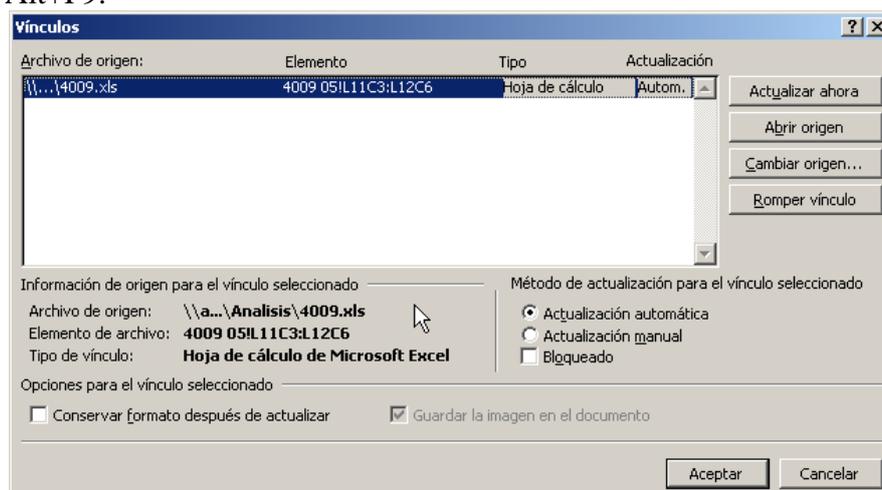


Otras posibilidades hubieran sido /Ventas/Datos si el fichero de conexión estuviera en esa ruta.

Esto no me ha funcionado.

4.3 Problema del office con conversión.

Al final después de mucho ver cuál era el problema, si trabajamos con una versión de Word anterior a la del 2007 y hacemos ahí las vinculaciones, al cambiar de directorio estas se actualizan correctamente y por tanto funcionan, pero si trabajamos con una versión del Word de 2007 aunque la convertimos posteriormente a compatibilidad 2003 o inferior los vínculos se convierten en absolutos y por tanto las actualizaciones las tendremos que hacer manualmente a través de la gestión de vínculos o modificándolos a través de Alt+F9.



Nos muestra el código del campo incrustado Alt+F9 : nos permite ver y modificar los códigos.

F9: Actualiza los campos

5 Hacer un sistema de ayuda para nuestra aplicación.

http://www.elguille.info/hhw/dnm/hhw_01.aspx

<http://nachocabanes.blogspot.com/2008/03/crear-ficheros-de-ayuda-de-windows-chm.html>

5.1 Preparativos del fichero ayuda en Word y generación de la Web.

En un documento Word (Manual Original) preparamos todo nuestro manual con imágenes y enlaces necesarios. Dejamos una carpeta preparada también que llamaremos Web donde generaremos con el FrontPage, DreamWeaver u otra aplicación similar para crear una web sencilla basada en la información contenida en el documento Word.

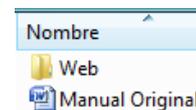


Ilustración 7

De esta forma con la información contenida en nuestro manual elaboramos una página web sencilla con la estructura que queramos dar a nuestro fichero final CHM (formato tipo de los ficheros de ayuda de Windows).

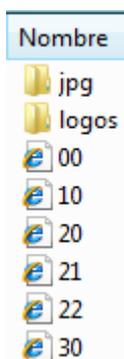


Ilustración 8

Como podemos observar en la Ilustración 8 hemos creado en la carpeta web una estructura de navegación con imágenes (carpeta jpg) y logos en los que se basa nuestro sistema de ayuda, así por ejemplo el fichero 00.html es el de presentación.



Ilustración 9

Cada uno de estos son ficheros independientes que contiene contenidos específicos de los que hemos desarrollado en el manual original (documento Word). Por tanto la base de nuestro proyecto de ayuda será esta página web que hemos desarrollado, evidentemente con todo su contenido como son las imágenes y enlaces. Una recomendación es el evitar ficheros bmp e ico, los demás formatos y en especial jpg son los recomendados.

De esta forma tendremos preparado nuestro manual correctamente en html y dispuesto a generar el fichero de ayuda CHM.

5.2 Creación de nuestro proyecto de forma simple.

5.2.1 Generación del fichero hhp.

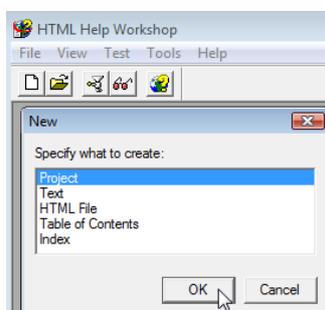


Ilustración 10

Es necesario ahora crear un proyecto de ayuda basado en la/s página/s web anteriormente generadas, para ello y en primer lugar cargamos la aplicación **hhw.exe (HTML Help Workshop)**, que es una pequeña utilidad gratuita de Microsoft, sino la tenemos la podemos descargar de la página de Microsoft.

Así por tanto abrimos la citada aplicación y seleccionamos la opción nuevo proyecto, con lo que nos aparece un asistente que nos guiará en esta parte, tal y como se muestra en las siguientes ilustraciones.

En el menú File (archivo), seleccionamos New (nuevo) y mostrará un cuadro de diálogo para indicar que tipo de archivo queremos crear, seleccionamos Project (proyecto).

A continuación nos mostrará un asistente, en la primera pantalla nos da la opción de convertir un fichero de ayuda tipo HLP (el usado antes de Windows 98), no marcamos esa casilla y pulsamos en siguiente (Next).

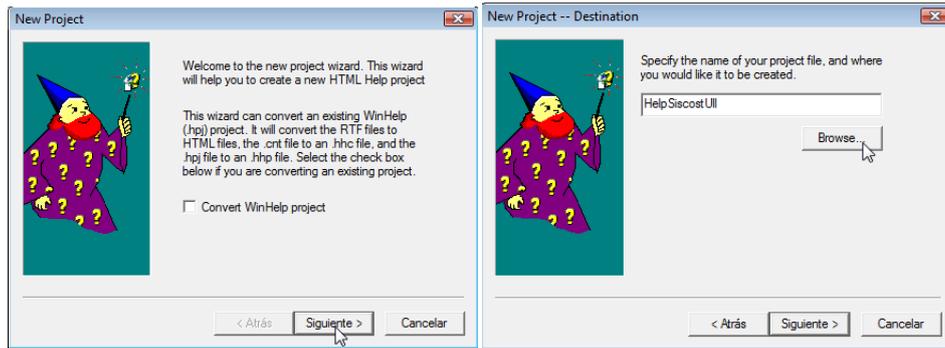


Ilustración 11

Posteriormente debemos indicar el nombre del fichero del proyecto, pulsamos en el botón Browse y seleccionamos el directorio en el que queremos crear el proyecto de ayuda, escribimos el nombre del proyecto y pulsamos aceptar y la ruta completa se mostrará en el asistente.



Ilustración 12

En nuestro caso lo hemos llamado al proyecto como HelpSiscostUII y el directorio donde lo almacenaremos será en la carpeta Web junto donde esta todo el documento Word convertido a htm.

En el siguiente paso nos pregunta si ya tenemos un fichero de contenido, de índice o ficheros HTML. Seleccionamos esa última opción para agregar los ficheros HTML que vamos a incluir.

A continuación nos solicita que indiquemos donde están localizados los ficheros html que son la base de nuestro sistema de ayuda y cuales queremos añadir al proyecto creado (HelpSiscostUII), ver Ilustración 13.

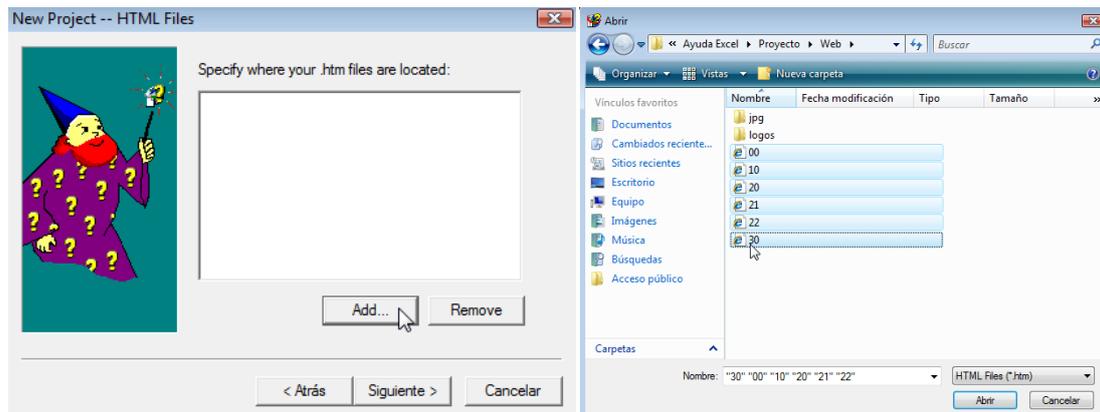


Ilustración 13

En nuestro caso seleccionamos todos los ficheros html creados en el apartado anterior para incorporarlos a nuestro proyecto y finalizamos esta primera etapa.

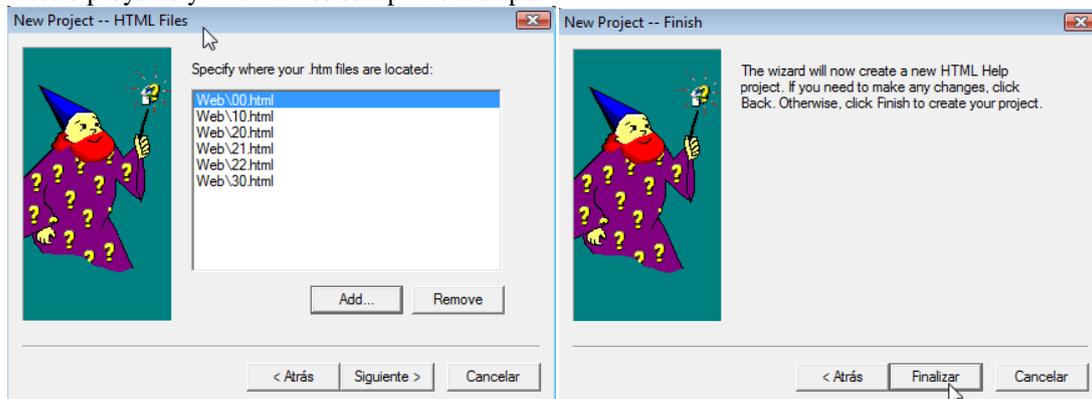


Ilustración 14

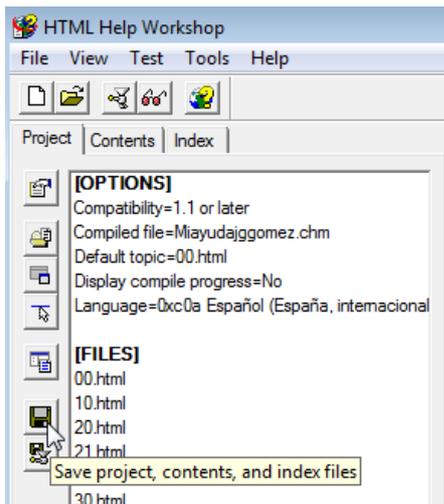


Ilustración 15

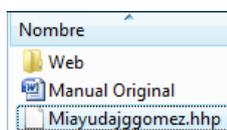


Ilustración 16

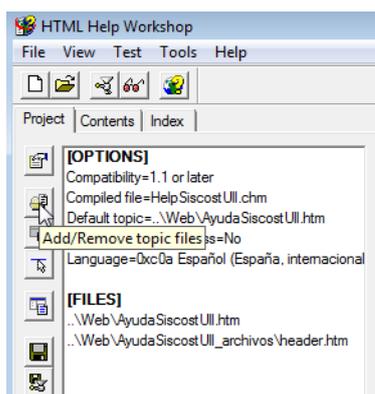


Ilustración 17

SEGUIR REVISANDO

5.2.2 Configurar las ventanas, crear la ventana Main

Una de las cosas que debemos hacer al crear un fichero de ayuda HTML es crear al menos una ventana. Como mínimo deberíamos crear una ventana, a la que le daremos el nombre "recomendado" de **Main**.

Para crear una ventana debemos pulsar en el tercer botón de la barra de herramientas de la izquierda (**Add/Modify window definitions**). Al pulsar en ese botón nos preguntará el nombre de la ventana, escribimos el nombre, (en nuestro caso será **Main**), y se mostrará otro cuadro de diálogo en el que tendremos que definir las características de esa ventana.

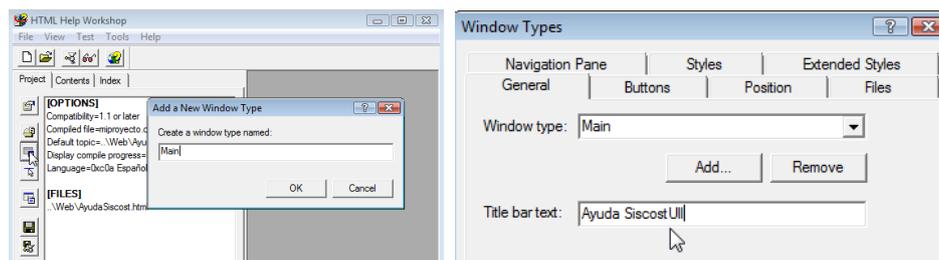


Ilustración 18

Para guardar el proyecto con el resultado del asistente, pulsaremos finalmente en el botón de guardar (el disquete) o bien en el menú **File>Save Project**. Lo podemos recuperar posteriormente para ampliarlo o modificarlo a través de la lectura del fichero generado anteriormente (open).

De esta forma una vez pulsado sobre el icono de la Ilustración 15 se habrá generado un fichero llamado tal y como se muestra en la Ilustración 16 (Fichero Miayudajggomez.hhp)

Como hemos manifestado posteriormente podemos agregar más ficheros HTML o eliminar algunos de los que hayamos agregado en este paso. Esto nos mostrará nuevamente la ventana principal de la utilidad, tal como vemos en la ilustración, en la que estarán los ficheros que hemos añadido.

Es decir, para agregar otros ficheros de contenido al proyecto de ayuda, en esta sesión o en otra posterior tenemos que seleccionar el segundo botón de la barra de herramientas que está a la izquierda de la ventana de la utilidad **HTML Help Workshop** (ver Ilustración 17).

Al pulsar en dicho botón podremos seleccionar los ficheros que queremos agregar o quitar a la ayuda.

Una vez que hemos agregado los ficheros HTML, estos se mostrarán bajo el marcador.

Nota:

Solo debemos agregar los ficheros HTML que vamos a usar, las imágenes y otros ficheros se incluirán de forma "automática", incluso los ficheros ZIP a los que tengamos links dentro de las páginas HTML.

En el cuadro de diálogo de definición de las ventanas, en la ficha **General** indicaremos el título a mostrar, en nuestro caso “Ayuda Siscost”.

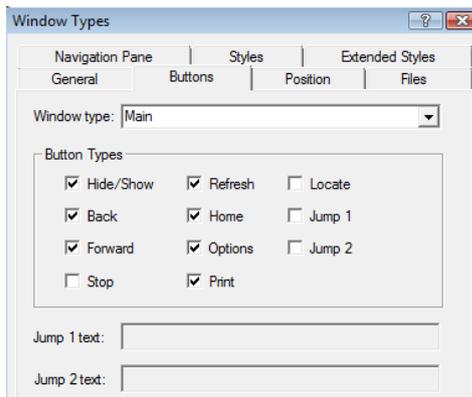


Ilustración 19

En la ficha **Buttons** indicamos los botones que queremos que se muestren en esta ventana. Si marcamos los botones **Jump 1** y/o **Jump 2**, tendremos que indicar el texto que se mostrarán en esos dos botones personalizados.

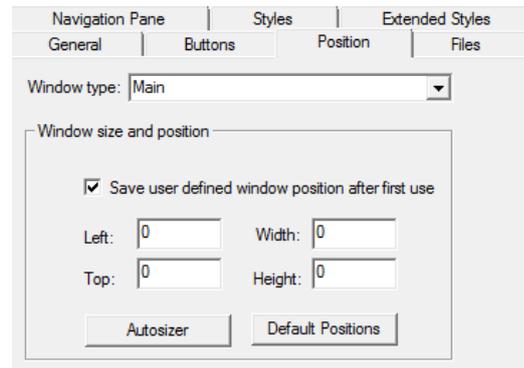


Ilustración 20

En la ficha **Position** indicamos el tamaño y la posición de esta ventana, o dejamos que sea el propio sistema el que se encargue de esos valores. Normalmente en la ventana principal (o por defecto), deberíamos dejar los valores predeterminados, aunque si queremos que la ayuda se inicie con un tamaño y en una posición de nuestra elección, podemos configurarlo.

También podemos marcar la opción de guardar los valores que el usuario le haya dado, en ese caso, debemos marcar la casilla *Save user defined window position after first use*. Aunque la posibilidad de que se recuerde la posición y tamaño que el usuario le da a la ventana, siempre estará disponible, incluso si no marcamos esa opción, al menos si dejamos los valores predeterminados (cero en los cuatro), en otro caso, al no marcar esa casilla, la ayuda siempre se mostrará en la posición y con el tamaño que indiquemos.

Si pulsamos en el botón *Autosizer*, se mostrará una ventana, el tamaño y posición que le demos a esa ventana serán los valores que se mostrarán en las cuatro casillas que indican la posición y tamaño de la ventana.

La ficha **Files** nos permite indicar los ficheros de contenido (**TOC**), índice (**Index**), la página HTML por defecto que se mostrará (**Default**) y los ficheros que se usarán si se pulsa en el botón **Home** y los dos opcionales para navegar a las dos páginas que indiquemos en **Jump 1** y **Jump 2** (para que se usen con los dos botones personalizados).

Todas esas opciones de ficheros son listas desplegables en las que nos mostrará los ficheros que podemos seleccionar en cada una de ellas. En los dos primeros (**TOC** e **Index**) nos mostrará solo los ficheros de contenido y de índice que tengamos creados, mientras que en el resto se mostrarán los ficheros HTML que hayamos agregado al proyecto de ayuda.

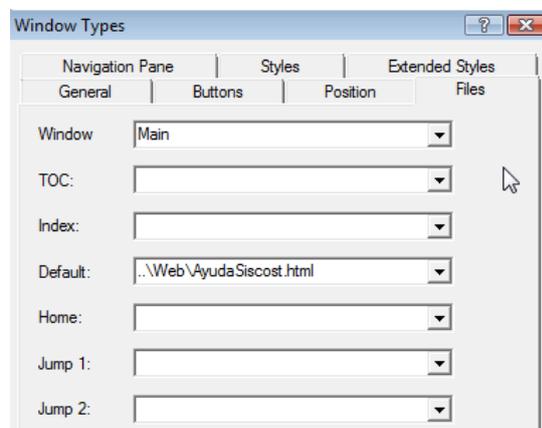


Ilustración 21

5.2.3 Compilando el fichero hhp, generando el fichero CHM Simple.

Ya podríamos ejecutar nuestra ayuda, compilando simplemente el fichero hhp para ello pulsamos sobre el botón "Compile HTML file". Eso es todo (si no queremos índices ni otras características avanzadas, claro).

Un ejemplo de fichero, que crearíamos con cualquier editor de texto (incluso el "Bloc de notas" serviría) y que guardaríamos con el nombre "ejemplo.hhp" (el nombre puede ser cualquiera, pero la extensión HHP es importante).

```
[OPTIONS]
Compatibility=1.1 or later
Compiled file=ejemplo.chm
Default topic=index.htm
Display compile progress=No
Language=0xc0a Español (España)
Title=Ejemplo de fichero de ayuda
[FILES]
index.htm
```

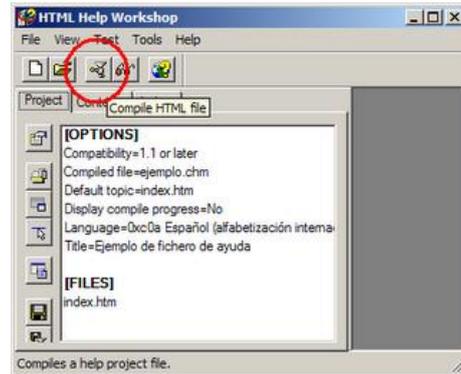


Ilustración 22

En muy pocos segundos tendremos un único fichero comprimido que agrupa toda nuestra estructura web, incluso imágenes.

5.3 Mejorando y ampliando nuestro proyecto.

5.3.1 Crear el contenido de la ayuda. Lista de Tópicos. *Fichero TOC.*

Una de las cosas que debemos hacer es crear el fichero de contenido, es decir, la lista de tópicos que se mostrarán en el panel de navegación.

Para crear un fichero de contenido, tendremos que pulsar en la ficha Contents de la ventana principal de HTML Help Workshop, al hacerlo, nos preguntará si queremos crear un fichero de contenido o usar uno existente .

Si pulsamos en *Create a new contents file*, nos mostrará un cuadro de diálogo para que indiquemos dónde lo queremos guardar, y si ya hemos estado trabajando con otros proyectos de ayuda, se usará el último directorio usado. En cualquier caso, lo mejor es seleccionar el mismo directorio en el que tenemos el fichero del proyecto de ayuda HTML.

Si ya tuviéramos un fichero creado, simplemente le decimos dónde está y se usará ese fichero de contenido.

Pero como estamos creando un nuevo proyecto de ayuda HTML, creamos uno nuevo. Al crear un nuevo fichero de contenido, tendremos en blanco esa parte y así se mostrará, tal como vemos en la siguiente ilustración, ahora lo que tenemos que hacer es llenar esa ficha de contenido.

Al pulsar en el botón Insert a heading nos mostrará un cuadro de diálogo (Ilustración 25) en el que indicaremos el texto a mostrar y si queremos

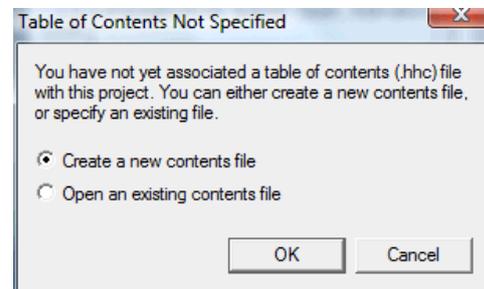


Ilustración 23

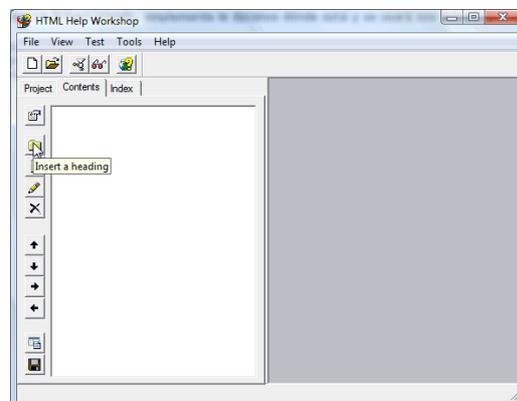


Ilustración 24. Crear nuevo encabezado

mostrar una página al pulsar en él.

Si decidimos usar una página a mostrar al pulsar en este "encabezado", lo podemos indicar pulsando en el botón Add, aunque por ahora no agregaremos ficheros de contenido a los apartados, esto lo haremos con los elementos que se mostrarán en cada sección.

Pulsando en la ficha Advanced podemos indicar la ventana que usaremos para mostrar ese contenido, si no indicamos nada, se usará la ventana principal (la que hemos definido como Main). Por ahora lo dejamos así, después veremos las opciones de esa ficha "avanzada".

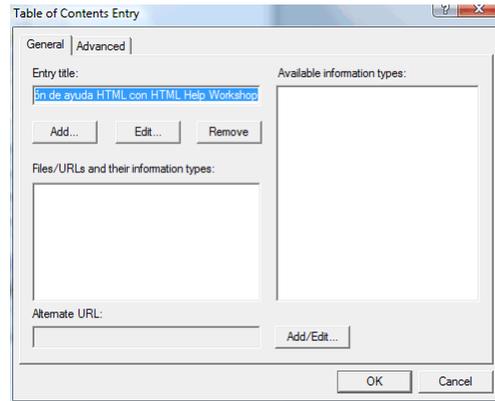


Ilustración 25. Opciones del nuevo elemento de Contenido

Nota:

En las entradas de encabezados no se suele "ligar" con ninguna página en concreto, pero podemos hacerlo sin problemas, de esa forma, cuando el usuario pulse sobre el encabezado se mostrará una página de contenido. Por ejemplo, es útil cuando un tópico se subdivide en otros tópicos que están relacionados.

Para agregar elementos de contenido, tenemos que pulsar en el tercer botón de la barra de herramientas (Insert a page), que es el icono con una hoja que está debajo de la carpeta que usamos en el paso anterior.

Cuando pulsamos en ese icono y tenemos el primer elemento seleccionado, nos muestra un aviso de si queremos agregarlo al principio o no (Ilustración 26). En este caso, pulsamos que no, para que el contenido se muestre debajo del apartado que creamos antes.

El cuadro de diálogo que nos muestra al pulsar en **Insert a page** es el mismo que vimos en la Ilustración 25. Si pulsamos en el botón **Add** nos muestra un nuevo cuadro de diálogo con las páginas HTML que tenemos en el proyecto de ayuda y debemos seleccionar la que nos interese mostrar al pulsar en esa página, y como vemos en la Ilustración 28, podemos indicar un marcador dentro de la posición.

Cuando pulsamos en aceptar (**OK**) se mostrará nuevamente el cuadro de diálogo en el que podemos seguir añadiendo más páginas o marcadores en esa misma u otra página. También podemos modificar esa entrada o algunas de las que hayamos añadido o quitarlas. Aunque, tal como vemos en la figura 7, lo habitual será que cada "hoja" solo tenga una dirección a la que navegar.

Nota:

Tal como acabo de comentar, podemos agregar varios links a una misma entrada, para saber cómo hacerlo, ver Agregar varios tópicos a una entrada en el siguiente capítulo.

Aunque esa "característica" es más habitual usarla en las entradas del índice que en la de contenido.

Nota:

Si el contenido de la ayuda lo creamos después de definir la ventana principal de la ayuda, debemos modificar dicha ventana de forma que indiquemos cual es el fichero de contenido.

La ficha Advanced la usaremos para indicar aspectos

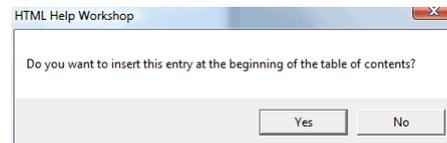


Ilustración 27. Aviso si queremos agregarle al principio.

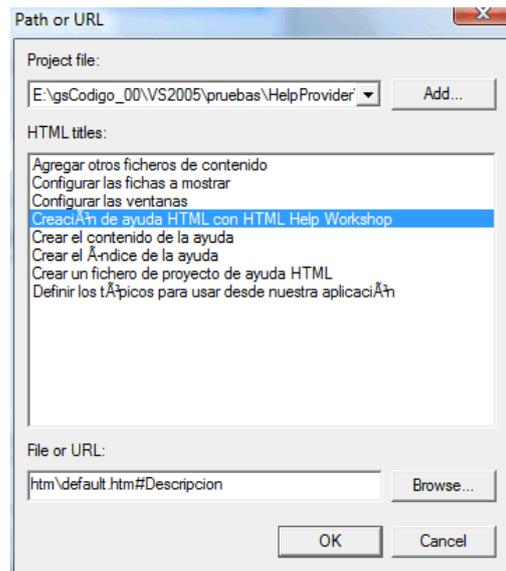


Ilustración 28. Indicar la página a la que irá el tópico agregado

peculiares de cómo y dónde debe mostrarse el elemento que hemos indicado. Tal como vemos en la Ilustración 30, en la ficha avanzada podemos indicar en que ventana se mostrará, si no indicamos ninguna, se usará la ventana principal que hayamos definido o la que el propio HTML Help Workshop define.

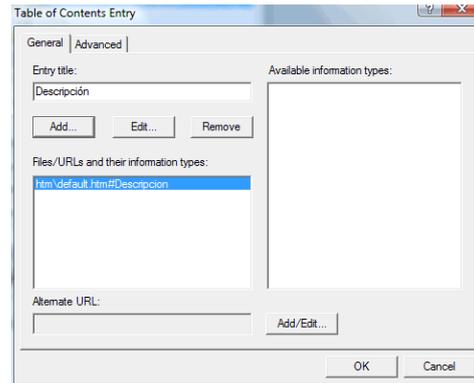


Ilustración 29. Editar una entrada en el contenido

En caso de que queramos usar una ventana especial, lo indicaremos en el campo Window. De la misma forma, si la ventana tiene definido algún "frame" (marco), lo indicaremos en el campo Frame. Si queremos resaltar la entrada como un nuevo elemento (se mostrará un asterisco rojo junto a la entrada), marcaremos la casilla Mark as new entry (ver la Ilustración 31). Que nos interesa que en vez de ser un elemento normal sea una entrada de cabecera (mostrada con un libro o una carpeta, según el tipo de icono elegido en la configuración general), marcaremos la opción Change entry to heading.

Por último, podemos elegir el icono a mostrar, eso lo haremos seleccionando el índice de la imagen (Image index), que por defecto está en auto con idea de que se utilice el icono correspondiente al tipo de entrada. En total hay 42 iconos entre los que elegir.

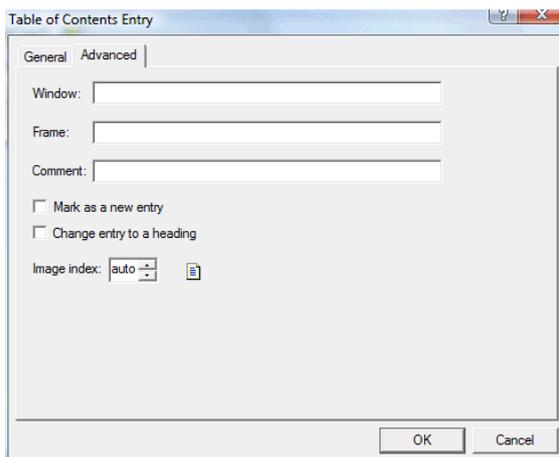


Ilustración 30. Ficha avanzada de configuración de tópicos

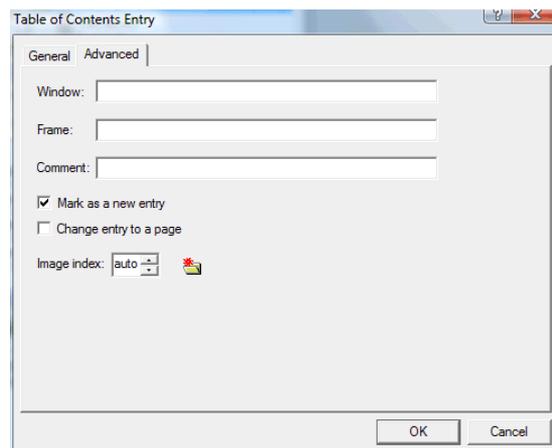


Ilustración 31. Configuración avanzada de una cabecera

Nota:

Al agregar una nueva entrada, podemos definir la ventana en la que se mostrará, pero "justo" al añadir esa entrada no podremos indicar el icono a mostrar, para poder especificar el icono, debemos agregar la entrada y después modificarla, para modificar una entrada, debemos seleccionar el tópico a modificar y pulsar en el botón con el icono del lápiz, tal como vemos en la Ilustración 32.

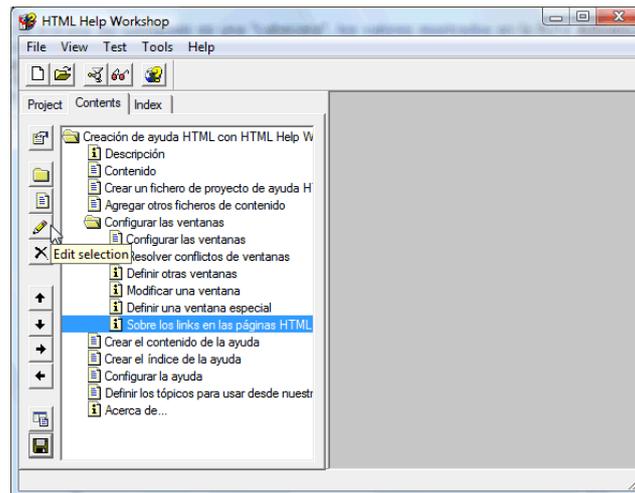


Ilustración 32. Editar una entrada, Tópico

5.3.2 Crear el índice de la ayuda. *Fichero Index.*

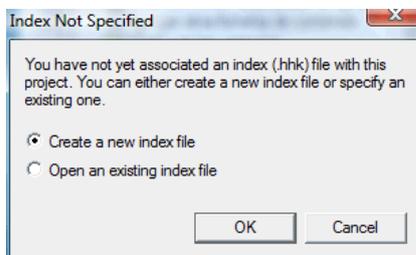


Ilustración 33. Crear o Usar un Fichero Índice

Si decidimos mostrar el índice de la ayuda, tendremos que crear los elementos que se mostrarán en esa ficha. Para crear el índice debemos pulsar en la ficha Index de la ventana principal de HTML Help Workshop, al hacerlo, tal como vemos en la Ilustración 33, nos preguntará si queremos usar un fichero existente o crear uno nuevo.

Como no tenemos ninguno creado, seleccionaremos la primera opción. Después de aceptar, nos preguntará por el nombre y la ruta donde queremos crear el fichero, el nombre por defecto del fichero con el índice de la ayuda es: Index.hhk.

Para agregar entradas al índice, usaremos el icono con la imagen de la llave (**Insert a keyword**), (Ilustración 34). El proceso a seguir es como vimos en el apartado anterior para agregar el contenido.

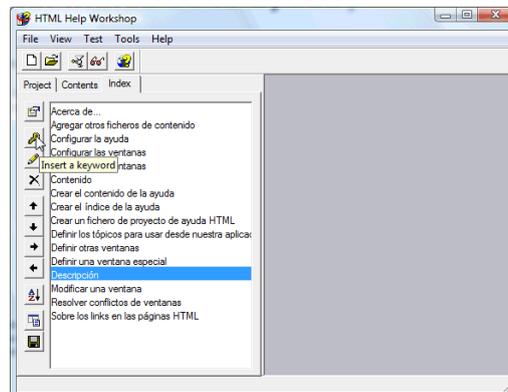


Ilustración 34. Las opciones de la ficha para el índice

Lo que debemos tener en cuenta es que el índice suelen se palabras clave que queremos que estén en la ayuda para que al usuario le resulte fácil encontrar diferentes "conceptos".

Truco: Aprovechar los tópicos del contenido para el índice

Debido que al principio el índice está vacío, yo lo que suelo hacer es cerrar el **HTML Help Workshop**, y hacer una copia del fichero de contenido (extensión **.hhc**), después le cambio el nombre para que tenga el del índice (**Index.hhk**) y así tengo algunas entradas ya incluidas, tal como se puede ver en la Ilustración 34.

Lo que hay que hacer con esas "entradas" del índice es quitarle la indentación (pulsando en la flecha que señala a la izquierda en cada uno de los elementos del índice) y clasificar el contenido del índice. Esto último lo haremos pulsando en el icono con **A/Z**.

Después voy agregando los elementos que quiera que estén en el contenido.

Agregar varios tópicos a una entrada

Tanto en la pestaña de contenido como en la del índice, podemos agregar a una misma entrada más de una página de contenido.

Esto lo conseguimos agregando una nueva entrada, asignamos el título a mostrar y pulsamos en el botón **Add** para añadir la página que queremos mostrar.

Si queremos añadir más páginas a una misma entrada, simplemente pulsamos en el botón **Add** tantas veces como queramos, agregando en cada ocasión un nuevo tópico a mostrar.

Por ejemplo, en la entrada **Ventanas** de la ayuda que estamos creando voy a añadir varias entradas, con idea de que el usuario elija entre las distintas opciones.

Al añadir esa entrada en el índice he añadido varias entradas, tal como vemos en la Ilustración 35. Cada una de esas entradas pueden tener su título correspondiente, ya que, por ejemplo, si son links a marcadores de una misma página, la utilidad siempre mostrará el título de la página principal.

Al seleccionar Ventanas en el índice y pulsar en el botón "Display" (o haciendo doble clic en la entrada), nos mostrará una ventana con las diferentes opciones que tenemos, tal como vemos en la Ilustración 36. Al pulsar en una de ellas, se mostrará el tópico que hemos asociado con ese texto.

Nota:

Si el índice lo creamos después de haber creado la definición de la ventana principal y queremos que se muestre la ficha de índice (Index), debemos modificar la ventana para que tenga la información del fichero de índice a usar.

En la Ilustración 37 podemos ver la ficha Files de la definición de la ventana principal (Main) con todos los ficheros que deben usarse en esa ventana. Al incluir tanto el de contenido como el del índice, se agregarán las fichas Contents (TOC) e Index a la definición de la ventana.

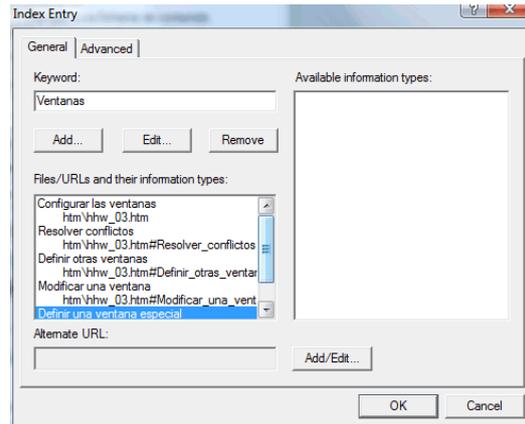


Ilustración 35. Agregar varias entradas a un mismo elemento.

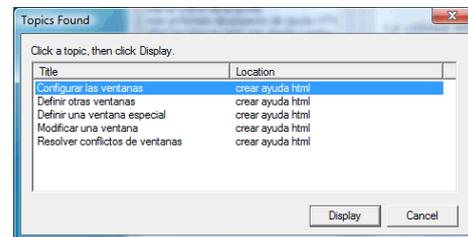


Ilustración 36. La lista de entradas de un mismo elemento del índice

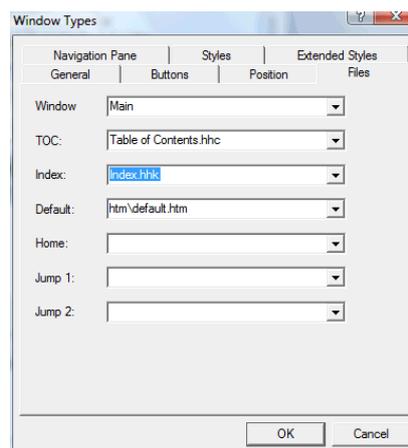


Ilustración 37. Debemos asegurarnos de que la ventana principal tenga todas las fichas que queremos mostrar

5.4 Modificar nuestro proyecto. Cambiar las opciones del proyecto.

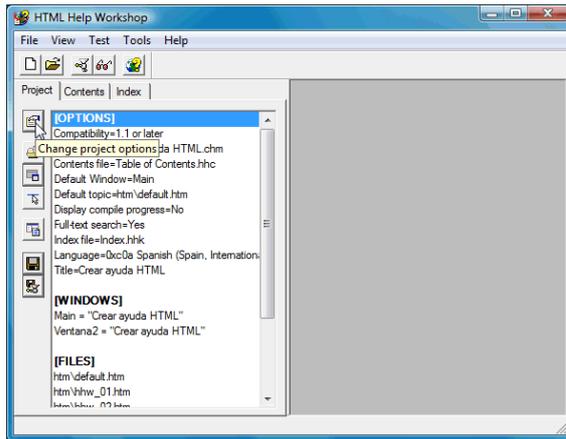


Ilustración 38. Cambiar las opciones del proyecto

Para configurar de forma general la ayuda, pulsaremos en el primer botón de la pantalla principal, el que muestra en el "tooltip" el texto Change project options (ver Ilustración 38).

Al pulsar en ese botón, nos mostrará un cuadro de diálogo con varias fichas, en la primera (General), tal como vemos en la

Ilustración 39, indicaremos el título de la ayuda, cual será la página por defecto (la que se mostrará al iniciar la ayuda) y la ventana que se usará cuando no se indique ninguna en particular.

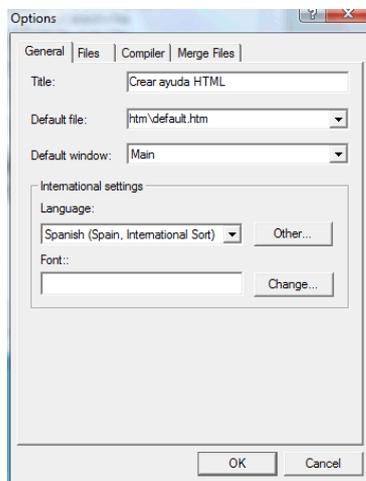


Ilustración 39. Ficha General de las propiedades del proyecto

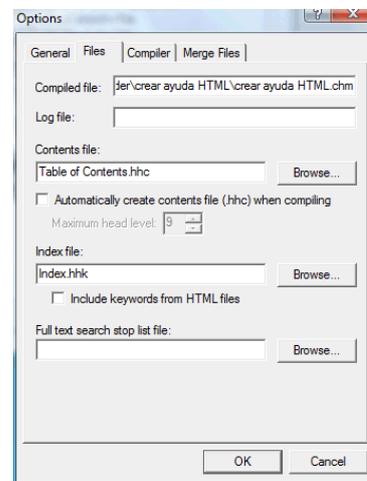


Ilustración 40. Ficha Files de las propiedades del proyecto

En la ficha Files (Ilustración 40), indicaremos el nombre y la ruta del fichero .chm, además de indicar cuales serán los ficheros que se usarán para el contenido (Contents file) y para el índice (Index file).

Si marcamos la opción Include keywords from HTML files (está bajo el nombre del fichero del índice), esto hará que se añadan automáticamente al índice las palabras clave que hayamos incluido en el código de los ficheros HTML.

Esas palabras clave no se incluyen en el "tag" keywords del encabezado (head) de la página, sino en objetos especiales que el sistema de ayuda usará para saber a dónde debe apuntar cuando se seleccione esa palabra clave.

Por último, en la ficha Compiler (ver la Ilustración 41), indicaremos que compatibilidad queremos (se recomienda usar la indicada de 1.1 o posterior) y otras opciones de compilación que podemos dejar con los valores que tienen por defecto, tal como vemos en la captura de la Ilustración 41.

Nota:

En el fichero de ayuda compilada (.chm) se mostrará aquí (debajo) un botón con links relacionados. Pero que está quitado del tutorial publicado en mi sitio, ya que te avisaría de que ese control se quiere ejecutar, etc.

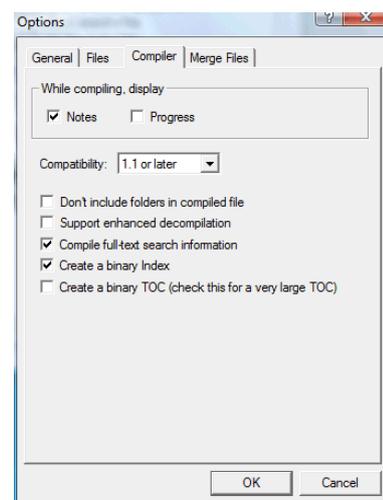


Ilustración 41. Ficha Compiler de las propiedades del proyecto

5.5 Definir los tópicos para usar desde nuestra aplicación. Configurar los alias de la ayuda

Para crear los tópicos debemos hacer lo siguiente en la utilidad HTML Help Workshop. En la barra de botones de la derecha de la utilidad debemos pulsar en el cuarto botón (HmlHelp API Information), tal como se muestra en la Ilustración 42. Esto nos mostrará un cuadro de diálogo con varias fichas tal como se muestra en la Ilustración 43, seleccionamos la segunda (Alias) y pulsamos en el botón Add.

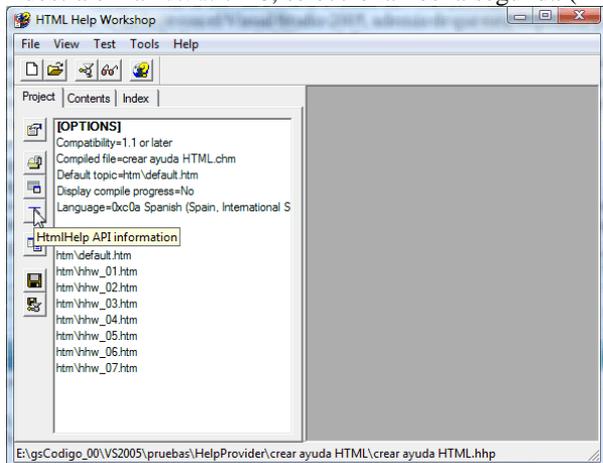


Ilustración 42. Configurar los alias de la ayuda

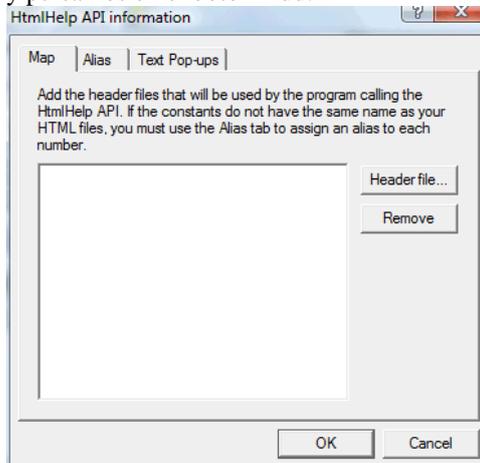


Ilustración 43. Opciones de HtmlHel API Information

Nos pedirá que indiquemos una constante o un número el cual se usará para hacer referencia a una página HTML de las que tenemos agregadas al proyecto (ver Ilustración 44).

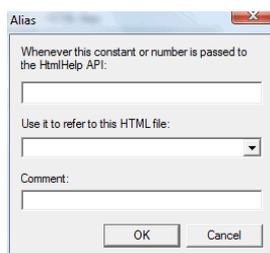


Ilustración 44. Añadir un alias a una página de contenido

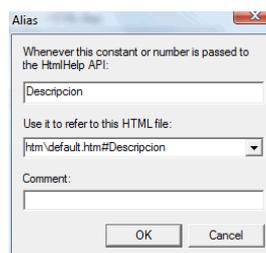


Ilustración 45. Las referencias pueden contener marcadores

Además de una página, podemos indicar un marcador dentro de una página, lo cual nos permite tener varias asociaciones dentro de una misma página HTML tal como vemos en la Ilustración 45

5.6 Programando el botón que llame al fichero de ayuda CHM.

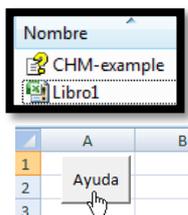
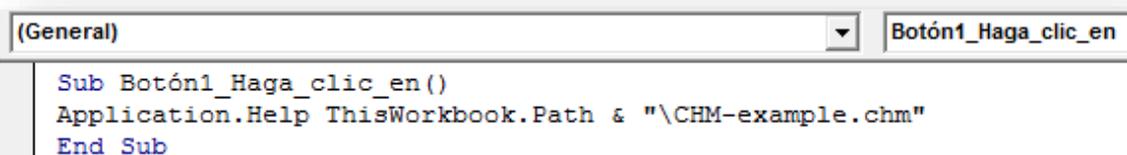


Ilustración 46

Nuestro objetivo es crear un sistema de ayuda para nuestra aplicación en concreto lo que deseamos es que en el libro 1 tenemos un botón que debe llamar al fichero tipo CHM que hemos preparado de ayuda para nuestra aplicación y que se encuentra en el mismo directorio.

Para ello basta con programar el botón ayuda con una simple macro tal y como se muestra a continuación:



6 Una cinta de opciones personalizada. Menú RibbonX.

6.1 Objetivo. Crear una barra de menú personalizada.

En este apartado aprenderemos a diseñar una barra de menú personalizada para nuestras aplicaciones desarrolladas en Excel, es decir algo similar a lo que mostramos en la Ilustración 47.

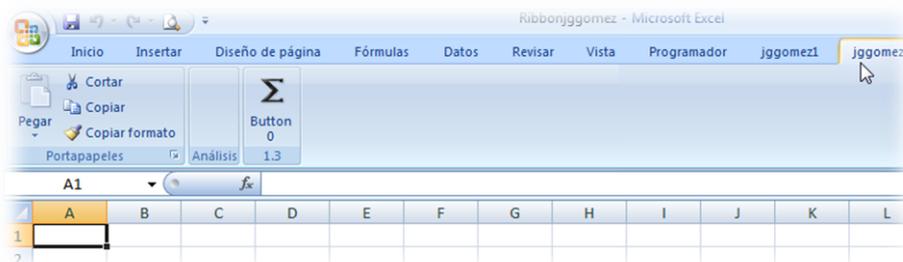


Ilustración 47

Con el fin de facilitar su desarrollo y adaptación haremos uso de una serie de herramientas que nos facilitarán su diseño y aplicación, sin embargo previamente mostraremos una serie de conceptos relacionados con esta cuestión.

6.2 Aspectos generales de la nueva barra de herramientas del office 2007.

6.2.1 La Cinta de opciones Ribbon.

En primer lugar debemos tener en cuenta que una cinta de opciones (Ribbon en inglés) es un elemento de las interfaces gráficas que consta de una tira o franja con múltiples funciones que el programa puede realizar. Por lo general, los ribbons suelen ubicarse en la parte superior de la ventana, aunque hay programas en donde se permite personalizar su ubicación y apariencia.

El objetivo del ribbon es ubicar todas las funcionalidades de una aplicación en un solo lugar para que sean fáciles de alcanzar. Por ejemplo, Microsoft Office 2007 emplea un ribbon llamado "Office Fluent Ribbon", que reemplaza los menús y las barras de herramientas.

Analicemos sus principales componentes para lo cual nos basaremos en la Ilustración 48 e **¡Error! No se encuentra el origen de la referencia.:**

1. **Cinta** La zona grande y rectangular que hay encima del documento se conoce como cinta. Contiene la barra de título, el botón de Office, la barra de herramientas de acceso rápido y las fichas. RibbonX se aplica principalmente a la cinta y a todo lo que hay en su interior.
2. **Botón de Office** Este botón abre el menú Office, el equivalente al menú Archivo de las versiones anteriores de Office. El menú Office contiene comandos que actúan sobre los documentos y no sobre el contenido de éstos. Los complementos de RibbonX pueden alterar libremente el contenido del menú Office (aunque no pueden personalizar el propio botón de Office).
3. **Barra de herramientas de acceso rápido** Esta barra de herramientas contiene comandos que se usan habitualmente y es la ubicación principal para las personalizaciones de los usuarios finales. Los usuarios pueden hacer clic con el botón secundario en cualquier control de la cinta y agregarlo a la barra de herramientas de acceso rápido (incluidos los controles personalizados de RibbonX). Como es el espacio que está pensado para que "pertenezca" al usuario final, normalmente los complementos de RibbonX no pueden alterar la barra de herramientas de acceso rápido a menos que tengan activado el modo StartFromScratch.
4. **Fichas (Tab)** Las fichas forman el contenido principal de la cinta y contienen controles de la interfaz de usuario que gestionan el contenido del documento actual. Los complementos de RibbonX pueden crear sus propias fichas personalizadas y alterar la visibilidad y las etiquetas de las fichas integradas.
5. **Conjuntos de fichas contextuales** Cuando se seleccionan objetos como imágenes o tablas dentro del documento, aparecen conjuntos de fichas contextuales que contienen todos los elementos de la interfaz de usuario para controlar dichos objetos. Los complementos de RibbonX pueden alterar la visibilidad de los conjuntos de fichas integrados y agregarles fichas

personalizadas. Una característica que no se admite en la versión 2007 de Office es la creación de conjuntos de fichas contextuales personalizados. Los conjuntos de fichas contienen fichas contextuales, que de otro modo se comportan igual que las fichas normales.

6. **Grupos (Groups)** Las fichas contienen conjuntos de grupos que, a su vez, contienen controles individuales de la interfaz de usuario. Los complementos de RibbonX pueden alterar la visibilidad de los grupos integrados y crear sus propios grupos personalizados. Algo que no pueden hacer es alterar el contenido de los grupos integrados. Esta limitación protege el diseño de la interfaz de usuario y evita que los complementos tengan conflictos entre sí y con futuras versiones de Office. Opcionalmente, los grupos tienen iniciadores de cuadros de diálogo en la esquina que muestran los diálogos relevantes para el grupo (como los diálogos de Fuente o Párrafo).

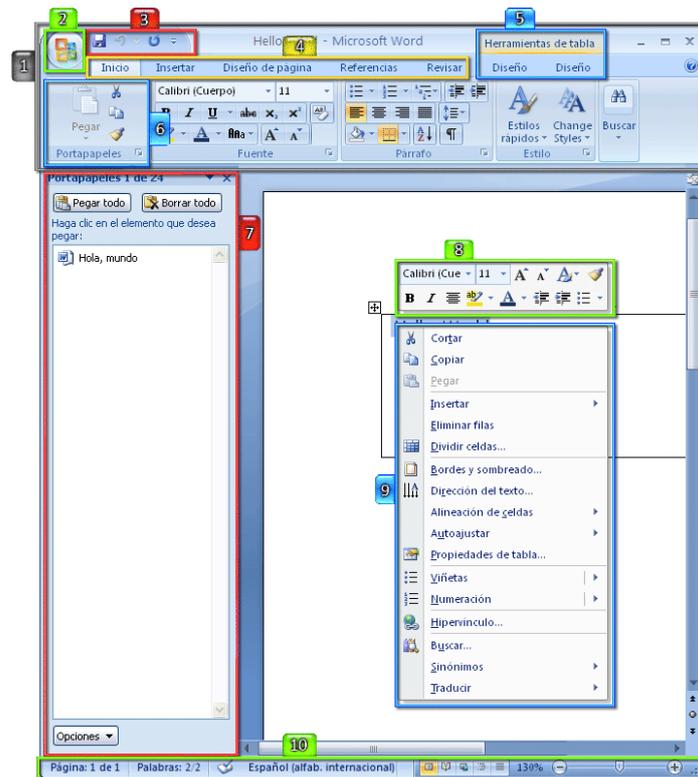


Ilustración 48



Ilustración 49

7. **Paneles de tareas** Hay muchos paneles de tareas que siguen presentes en el sistema Office 2007 y ahora se pueden tener abiertos más de uno a la vez. Los complementos de COM ahora pueden crear CustomTaskPanes que albergan contenido como controles ActiveX® o controles Windows® Forms. (La característica CustomTaskPane es distinta de RibbonX y no se trata en este artículo.)
8. **MiniToolbar** MiniToolbar es una colección de comandos de formato habituales que aparecen sobre las selecciones de texto y los menús contextuales que se abren al hacer clic con el botón

secundario. Los complementos de RibbonX no pueden modificar el contenido de MiniToolbar, pero pueden desactivar o volver a planificar los comandos en ella.

9. **Menús contextuales** Son los mismos menús contextuales que se abren al hacer clic con el botón secundario que todos conocemos y apreciamos de las versiones anteriores de Office. En la versión 2007 de Office, RibbonX no se aplica a los menús contextuales, pero se pueden ampliar y personalizar usando el modelo de objetos CommandBars igual que antes.
10. **Barra de estado** La barra de estado contiene varios controles nuevos y prácticos como el número de palabras y la vista del control deslizante. La barra de estado no se puede personalizar con los complementos en el sistema Office 2007, aunque se puede ocultar.

6.2.2 El formato XML de la cinta de opciones Ribbon.

La **nueva interfaz** de Office **Excel 2007** está realizada con los nuevos formatos [XML] de código abierto. Como su nombre indica **XML** (eXtensible Markup Language) ...traducido (lenguaje extensible de marcas) y que con este nuevo uso de **XML** en **Office 2007** nos permitirá adaptar la nueva Interfaz agregando nuevas Etiquetas, grupo de botones, combobox, popus y un largo etc.....

Así, por tanto podemos generar nuestra cinta de opciones en lenguaje XML es decir creando un fichero XML que asociaremos al documento office creado previamente.

Al respecto debemos tomar en consideración una serie de aspectos relacionados con este tipo de fichero XML, así y en primer lugar apuntar que todos los controles de la cinta de opciones deben incluir uno de los siguientes elementos identificadores:

- id: que especifica el identificador único del control y que se usa con controles personalizados
- idMso: que especifica el identificador de un menú propio de Office
- idQ: que especifica un identificador cualificado, precedido de una abreviatura del espacio de nombres.

Para usar en las personalizaciones los menús de Office debemos conocer sus "nombres internos". Microsoft proporciona una lista con todos esos nombres internos y que podéis descargar desde aquí desde su [página web](#).

En Access 2007, podemos personalizar la cinta de opciones creando primero código XML de personalización y, después, agregando código o estableciendo propiedades de base de datos que den a Access la instrucción de utilizar dicho código XML cuando cree la cinta de opciones. Podemos utilizar el código XML para ocultar fichas existentes y agregar fichas, grupos de comandos y comandos nuevos.

El código XML se puede almacenar en varias ubicaciones, pero uno de los métodos más sencillos es almacenarlo en una tabla del sistema de la base de datos activa. El proceso implica crear una tabla del sistema denominada USysRibbons, agregar a la tabla el código XML de la cinta de opciones y, a continuación, especificar si la cinta de opciones personalizada se ha de mostrar para toda la base de datos o para un formulario o informe específico.

Podemos definir varias cintas de opciones personalizadas, una para toda la aplicación y las demás para formularios o informes individuales de la base de datos.

Igualmente en Excel también podemos personalizar la cinta de opciones siguiendo la misma metodología.

6.2.3 Programas de apoyo para el desarrollo de la cinta de opciones. Ribbon Creator y Custom UI Editor.

Para ayudar al diseño de la cinta de opciones Ribbon y facilitar nuestro desarrollo nos hemos apoyado en dos herramientas básicas y que estudiaremos a continuación como son: Ribbon Creator (<http://www.ribboncreator.de/en/>) y el Office 2007 Custom UI Editor (<http://openxmldeveloper.org/articles/customuieditor.aspx>).

Con el primero generaremos y personalizaremos nuestra barra de opciones o cinta Ribbon y con el segundo intentaremos adaptarlo a una aplicación existente, fichero Word, Excel o Access.

6.3 Programa Ribbon Creator.

6.3.1 Creación de un Nuevo Proyecto Ribbon.

En este caso vamos a crear una cinta de menús personalizada para una nueva hoja de cálculo. Para ello y en primer lugar cargamos nuestra aplicación y comenzamos creando una nueva cinta de menú tal y como se muestra en la Ilustración 50.

De esta forma accedemos a la pantalla de configuración general de nuestra cinta de opciones (Ilustración 51).

Es este apartado donde podremos configurar y adaptarla a nuestras necesidades.

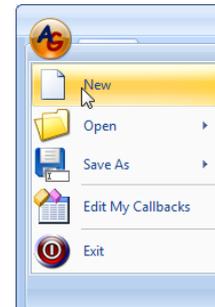


Ilustración 50

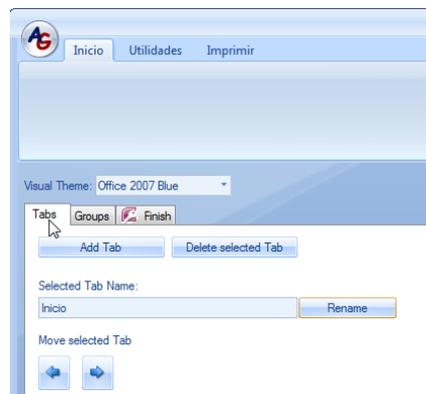


Ilustración 51

El apartado de configuración consta de tres apartados, la pestaña Tab (Ficha), Groups y Finish. En nuestro caso hemos comenzado por crear y diseñar básicamente tres opciones tabs o fichas que son Inicio, Utilidades e Imprimir (ver Ilustración 51).

Si quisiéramos en este momento guardar nuestro proyecto, podremos hacerlo de tres formas:

1. **Salvar con aplicación Ribbon** (ver Ilustración 51). En este caso crearemos un nuevo un documento Word o Excel con la nueva cinta de menú general que hemos diseñado y que se combinaría con la estándar preestablecida para este tipo de documento (ver Ilustración 53).

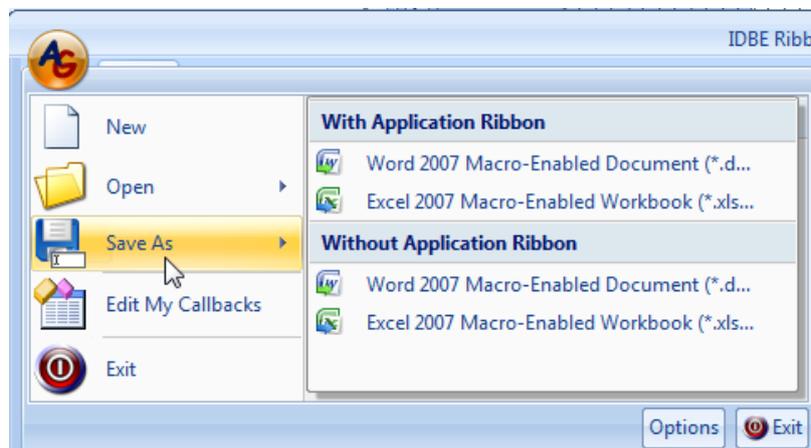


Ilustración 52

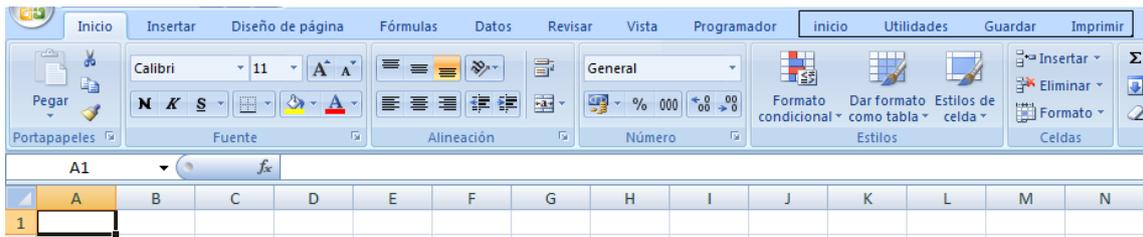


Ilustración 53

2. **Salvar sin aplicación Ribbon** (ver Ilustración 51). En este caso crearemos un nuevo un documento Word o Excel con la nueva cinta de menú general que hemos diseñado donde no contaremos con la cinta estándar preestablecida para este tipo de documento (ver Ilustración 54).

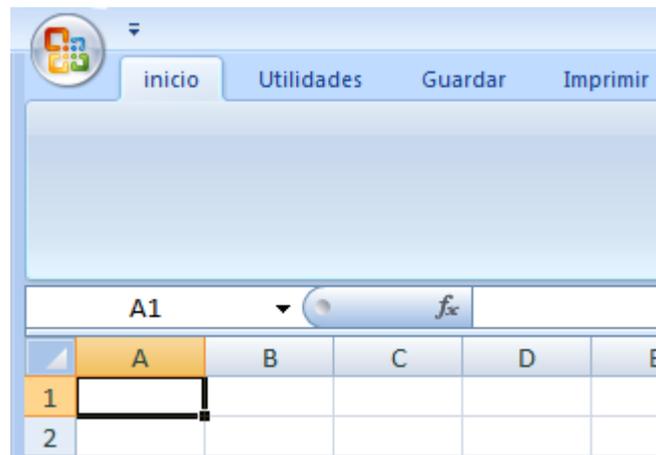


Ilustración 54

3. **Salvar como fichero XML** (ver Ilustración 55). En este caso crearemos un fichero XML con la configuración de la nueva cinta de menú que hemos diseñado. De esta se forma

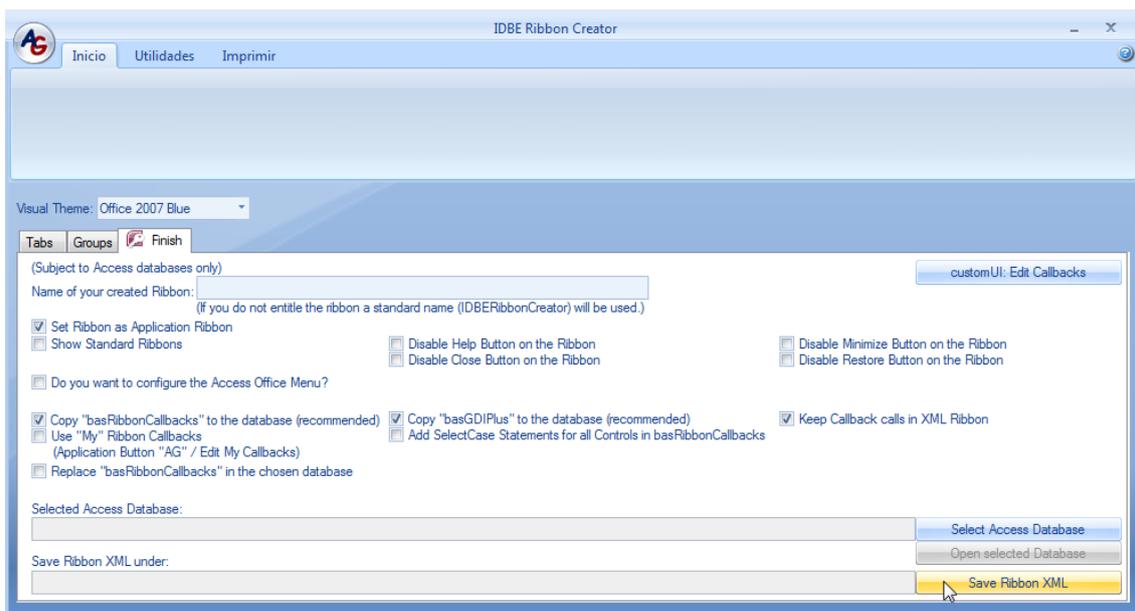


Ilustración 55

6.3.2 Crear un cinta de opciones con Ribbon Creator para aplicarla a un fichero Excel existente.

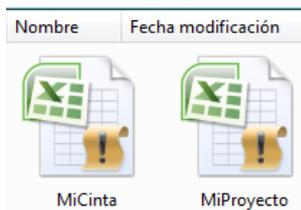


Ilustración 56

En este caso que planteamos vamos a crear una cinta de opciones para un proyecto que ya tenemos desarrollado en Excel y para ello en primer lugar crearemos nuestra barra de menús personalizado tal y como hemos detallado en el apartado anterior 6.3.1. esto consecuentemente nos generara un nuevo fichero Excel 2007 en formato libro Excel XML habilitado para macros y al que hemos llamado MiCinta.xml.

Por otro lado contamos con nuestro proyecto creado en Excel y que tiene el nombre de MiProyecto con el mismo formato que el anterior.

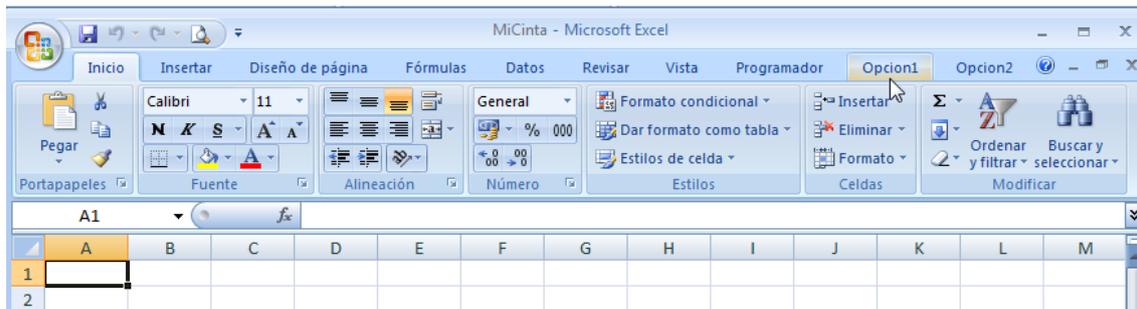


Ilustración 57

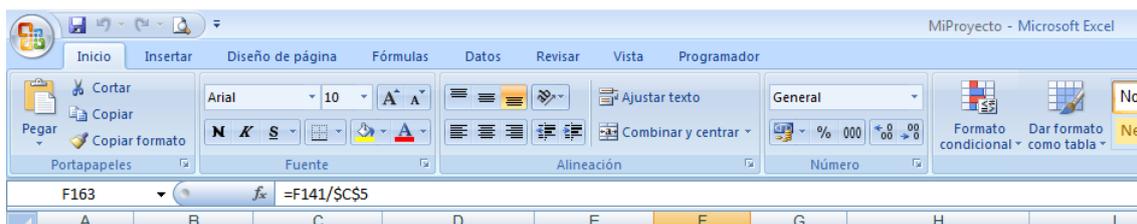


Ilustración 58

Como podemos ver de la Ilustración 57 y Ilustración 58, la hoja de cálculo MiCinta presenta una cinta de opciones personalizada con Ribbon Creator (opción1 y opción2) mientras que el fichero MiProyecto tiene una cinta de opciones generales. Por tanto nuestra intención es copiar o adaptar la cinta de opciones de la Ilustración 57 al fichero MiProyecto.

Para ello y en primer lugar vamos a ver el contenido en términos VBA de MiCinta, para ello entramos en la opción Programador y vemos el contenido VBA de ese fichero (MiCinta).

Podemos observar como la aplicación Ribbon Creator ha generado dos módulos y también, aunque no lo podamos visualizar ha insertado un fichero XML asociado a la citada cinta.

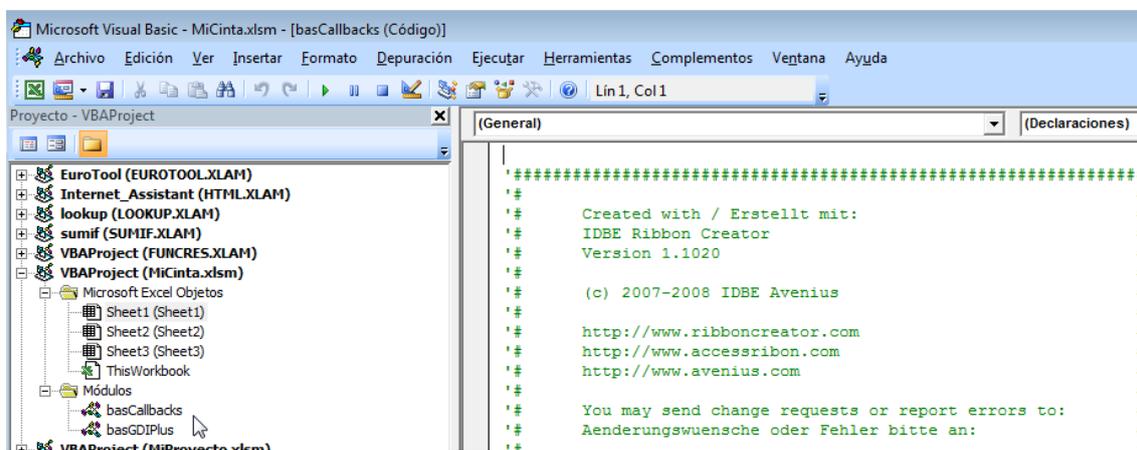


Ilustración 59

Por tanto basta con copiar los módulos al fichero deseado (arrastrando los dos módulos al proyecto MiProyecto) tal y como se muestra en a continuación.

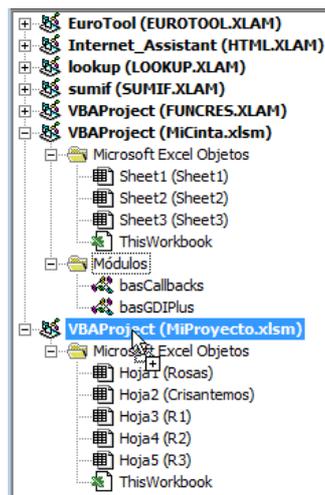


Ilustración 60

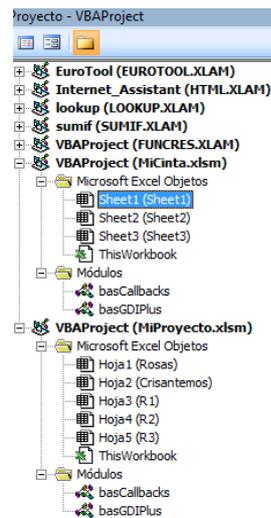


Ilustración 61

Pero como hemos comentado anteriormente es necesario contar además con el código XML que ha generado también el Ribbon Creator y aplicarlo o pegarlo en nuestra hoja de calculo. Para ello vamos a leer el citado código y prepararnos para copiarlo.

Para ello abrimos una herramienta que nos ayudara como es Custom UI Editor y con ella accedemos al código XML asociado al fichero MiCinta, ver Ilustración 62.

```

<!-- Created with IDBE Ribbon Creator (Version: 1.1020c) -->
<!--      http://www.RibbonCreator.com      -->
<!--      http://www.RibbonCreator.de      -->

<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" onLoad="OnRibbonLoad">
  <ribbon startFromScratch="false">
    <tabs>
      <tab id="tabOpcion1" label="Opcion1" getVisible ="GetVisible">
      </tab>
      <tab id="tabOpcion2" label="Opcion2" getVisible ="GetVisible">
      </tab>
    </tabs>
  </ribbon>
</customUI>

```

Ilustración 62

Copiamos este código en el notepad y abrimos con el mismo programa Custom UI Editor el XML asociado al fichero MiProyecto. Podemos ver que el mismo se encuentra vacío (Ilustración 63). Por tanto simplemente pegamos el código que tenemos cargado en memoria y de esta forma esta asociado ya nuestra cinta de opciones al proyecto

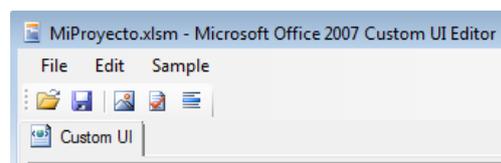


Ilustración 63

Al abrir ahora nuestro fichero MiProyecto el resultado sería el que se muestra en la Ilustración 64, es decir con la incorporación de nuestras opciones.



Ilustración 64

6.3.3 Vincular el complemento Excel a nuestra hoja de calculo personalizada.

Ahora al estar vinculado ya nuestra cinta de opciones al programa Ribbon Creator la podemos modificar y cambiar a nuestro gusto, simplemente abriendo desde la citada aplicación este fichero y haciendo las modificaciones oportunas.

7 Otras Cuestiones

7.1 Insertar imágenes en Excel como iconos, aplicar para menú

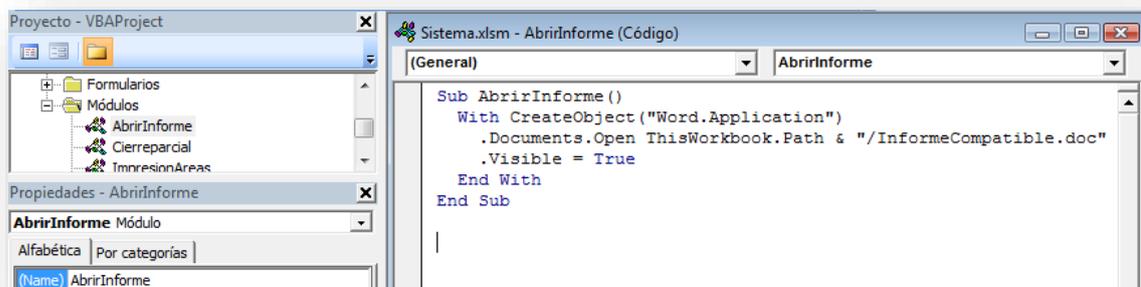
Para meter imágenes en Excel para utilizarlas como menú, recomendamos la de los Simpson copiar y pegar y después cambiarle el nombre

7.2 Hacer referencia a una ruta relativa en Excel.

Necesitamos hacer una referencia a una ruta relativa de un fichero desde Excel, por ejemplo para abrir un documento Word, la sentencia sería:

```
RutArchivo = thisworkbook.path & "/" "Ejemplo.xls"
thisworkbook.path & "/Ejemplo.xls"
```

Veamos un caso concreto, este será un modulo general que hemos creado para abrir un documento Word.



7.3 Abrir un documento Word desde Excel y situarlo en una página concreta.

El objetivo planteado en esta ocasión es el disponer un botón en una hoja Excel que nos abra un determinado documento Word. Para ello vamos a crear modulo normal donde introduciremos el siguiente código.

```
Sub AbrirDocumentoWord()
With CreateObject("Word.Application")
.Documents.Open "c:\ruta y\nombre del documento.doc"
.Visible = True
End With
End Sub
```

Si quisiéramos perfeccionarlo y que nos situara automáticamente en una pagina concreta del documento Word requerido, el código sería:

```

Sub AbrirDocWordEnPagina()
  With CreateObject("Word.Application")
    .Documents.Open ("C:\Ruta y\Nombre del archivo.doc")
    .Selection.GoTo What:=wdGoToPage, Which:=wdGoToNext, Name:="10"
    .Visible = True
  End With
End Sub

```

7.4 ¿Puedo hacer que una o varias celdas del libro parpadeen?

Sí, es posible hacerlo, pero hay que tener en cuenta que para conseguirlo es necesario ejecutar código casi continuamente, lo cual podría interferir con otro código que tuviera el libro. Lo mejor es usar el parpadeo lo menos posible, y sólo en libros que no tengan más código.

7.5 Macros Auto_Open() y Auto_Close().Workbook_Open

Existe una macro de autoarranque que se ejecuta cuando se abre el libro. Se llama auto_open() y existe otra que se ejecuta justo antes de cerrar el libro que se llama auto_close().

```

Sub Auto_Open()
  Dim hora As Double
  Dim saludo As String
  hora = (Now - Int(Now)) * 24
  Select Case hora
    Case 6 To 14
      saludo = "Buenos días"
    Case 14 To 21
      saludo = "Buenas tardes"
    Case Else
      saludo = "Buenas noches"
  End Select
  MsgBox saludo & " Amo"
End Sub

```

Equivalente a auto_open existe otra macro, pero ésta ha de ser guardada no en un módulo normal, sino en ThisWorkbook. Pruebe el siguiente procedimiento que permite abrir automáticamente el libro Balance.xls al abrir el libro Informe.xls. El procedimiento ha de estar en ThisWorkbook del libro Informe.xls.

```
Sub Workbook_Open()  
    'Apertura de libro Balance  
    Workbooks.Open Filename:="C:/Temp/Balance.xls"  
    'Activación del libro Informe  
    Windows("Informe.xls").Activate  
End Sub
```

8 Bibliografía.

<http://www.fermu.com/content/view/397/2/lang.es/>

<http://www.observatorio.cnice.mec.es> Programación Excel Observatorio Tecnológico
Aprendiendo Programación con Microsoft Excel 2000 en 24 Horas. Sharon Podlin. Editorial Pearson
Educación 2001.

http://www.elguille.info/hhw/dnm/hhw_01.aspx